



TUGAS AKHIR - TE 145561

***MONITORING PERHITUNGAN DAN PENGELOMPOKAN
BARANG DI KONVEYOR BERDASARKAN KODE BATANG
(SCANNER BARCODE) MELALUI LCD***

Aristawati Dhiyan Widodo
NRP 10311500010021

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TE 145561

***MONITORING CALCULATION AND CLASSIFY THE
AMOUNT OF OBJECTS IN CONVEYOR BASED ON
SCANNER BARCODE WITH LCD***

Aristawati Dhiyan Widodo
NRP 10311500010021

Supervisor
Ir. Josaphat Pramudijanto, M.Eng.

*Electrical Automation Engineering Department
Faculty of Vocational*
Institut Teknologi Sepuluh Nopember
Surabaya 2018

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Monitoring Perhitungan dan Pengelompokan Barang di Konveyor Berdasarkan Kode Batang (Scanner Barcode) Melalui LCD**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2018



Aristawati Dhyan Widodo
10311500010021

-----Halaman ini sengaja dikosongkan-----

**MONITORING PERHITUNGAN DAN PENGELOMPOKAN
BARANG DI KONVEYOR BERDASARKAN KODE BATANG
(BARCODE) MELALUI LCD**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Ahli Madya Teknik**

Pada

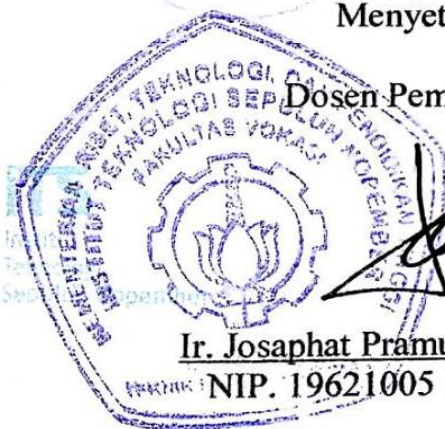
Departemen Teknik Elektro Otomasi

Fakultas Vokasi

Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing



Ir. Josaphat Pramudijanto, M. Eng

NIP. 19621005 199003 1 003

**SURABAYA
JULI, 2018**

-----Halaman ini sengaja dikosongkan-----

MONITORING PERHITUNGAN DAN PENGELOMPOKAN BARANG DI KONVEYOR BERDASARKAN KODE BATANG (SCANNER BARCODE) MELALUI LCD

Nama Mahasiswa : Aristawati Dhiyan Widodo
NRP : 10311500010021
Dosen Pembimbing : Ir. Josaphat Pramudijanto, M.Eng
NIP : 19621005 199003 1 003

ABSTRAK

Proses distribusi barang di industri berpengaruh dalam proses jual-beli ke masyarakat. Oleh karena itu, dibutuhkan suatu alat untuk membantu proses distribusi tersebut. Sehingga proses distribusi akan lebih menghemat waktu, meringankan pekerjaan manusia, dan dapat meminimalisir adanya *human error* yang sering terjadi di industri manual. Para pekerja juga hanya perlu memantau proses distribusi tersebut melalui tampilan LCD tanpa harus mengeceknya setiap saat.

Konveyor digunakan untuk mendistribusikan barang dari satu tempat ke tempat lainnya. *Scanner barcode* MCR12 diletakkan di bagian atas konveyor tersebut, sehingga barang yang berlabel *barcode* ITF-6 ketika melewati konveyor akan terdeteksi. Kemudian, *scanner barcode* akan mengirimkan data dan mengolahnya di program Arduino Mega 2560 R3. Arduino akan mengaktifkan *relay* sesuai program dan *relay* akan mengaktifkan motor pada konveyor, dan konveyor akan bekerja sesuai *barcode* yang terdeteksi.

Scanner barcode MCR12 ini dapat mendeteksi *barcode* dengan baik dalam jarak 20cm. Posisi yang tepat *scanner barcode* mendeteksi *barcode* adalah 90° dan 270° dalam rotasi horizontal. Keberhasilan *scanner barcode* dalam mendeteksi *barcode* dengan tepat yaitu 64%. Dan, waktu rata-rata proses pengelompokan *barcode* A, B, dan C adalah 18 detik, 32 detik, dan 16,7 detik. Sensor ini tidak dapat terlalu lama bekerja, sehingga seringkali harus memberi jeda agar *barcode* tetap dapat terdeteksi. Untuk konveyor, ketika pengelompokan barang seringkali barang terlalu cepat atau terlalu lambat berjalan.

Kata Kunci : Arduino, Barcode, Konveyor, Scanner Barcode

-----Halaman ini sengaja dikosongkan-----

MONITORING CALCULATION AND CLASSIFY THE AMOUNT OF OBJECTS IN CONVEYOR BASED ON SCANNER BARCODE USING LCD

Name of Student : Aristawati Dhiyan Widodo
Number of Registration : 10311500010021
Supervisor : Ir. Josapahat Pramudijanto, M.Eng
ID Number : 19621005 199003 1 003

ABSTRACT

The process of distribution of objects in the industry affects the process of buying and selling to the community. Therefore, a tool is needed to assist the distribution process. So that the distribution process will save time, ease the human work, and can minimize the human error that often occurs in the manual industry. The workers also only need to monitor the distribution process through the LCD display without having to check it at any time.

Conveyors are used to distribute objects from one place to another. The MCR12 barcode scanner is placed at the top of the conveyor, so that the item labeled ITF-6 barcode when it passes through the conveyor will be detected. Then, the barcode scanner will send the data and process it in Arduino Mega 2560 R3 program. The arduino will activate the relay according to the program and the relay will activate the motor on the conveyor, and the conveyor will work according to the detected barcode.

This MCR12 barcode scanner can detect barcodes well within 20cm. The exact position of barcode scanners detecting barcodes is 90° and 270° in horizontal rotation. The success of barcode scanners in accurately detecting barcodes is 64%. And, the average time of the process of grouping barcodes A, B, and C is 18 seconds, 32 seconds, and 16.7 seconds. This sensor can not be too long to work, so it often has to give pause to keep the barcode detected. For conveyors, when grouping of goods is often too fast or too slow to walk.

Key Words : Arduino, Conveyor, Scanner Barcode

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada junjungan kita Nabi Besar Muhammad SAW.

Tugas Akhir ini disusun untuk memenuhi sebagai persyaratan guna menyelesaikan pendidikan Diploma pada Bidang Studi Elektro Industri, Program Studi D3 Teknik Elektro, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul :

"Monitoring Perhitungan dan Pengelompokan Barang di Konveyor Berdasarkan Kode Batang (*Scanner Barcode*) Melalui LCD"

Dengan terselesaikannya Tugas Akhir ini, Penulis menyampaikan terima kasih yang sebesar - besarnya kepada :

1. Kedua orang tua beserta anggota keluarga lainnya yang senantiasa mendoakan dan memberikan dukungan dengan tulus tiada henti.
2. Bapak Ir. Josaphat Pramudijanto, M.Eng selaku dosen pembimbing.
3. Bapak Abdul Ghofur, S.T. selaku dosen pembimbing dari PT. Campina Ice Cream Industry Tbk.
4. Teman - teman Departemen Teknik Elektro Otomasi Hydra yang selalu memberikan doa, bantuan, semangat, dan dukungannya.
5. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat serta bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, Juli 2018

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

| | HALAMAN |
|---|---------|
| HALAMAN JUDUL | i |
| HALAMAN JUDUL | iii |
| PERNYATAAN KEASLIAN TUGAS AKHIR..... | v |
| HALAMAN PENGESAHAN | vii |
| ABSTRAK..... | ix |
| <i>ABSTRACT</i> | xi |
| KATA PENGANTAR | xi |
| DAFTAR ISI | xv |
| DAFTAR GAMBAR..... | xvii |
| DAFTAR TABEL | xix |
| BAB I PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Permasalahan | 2 |
| 1.3 Batasan Masalah | 2 |
| 1.4 Tujuan..... | 3 |
| 1.5 Sistematika Laporan | 3 |
| 1.6 Relevansi | 4 |
| BAB II TEORI PENUNJANG | 5 |
| 2.1 Konveyor | 5 |
| 2.1.1 <i>Belt Conveyor</i> | 5 |
| 2.1.2 <i>Plastic Flat Chain Conveyor</i> | 6 |
| 2.1.3 <i>Vibrating Conveyor</i> | 6 |
| 2.1.4 <i>Hanging atau Overhead Conveyor</i> | 7 |
| 2.2 <i>Relay</i> | 7 |
| 2.3 <i>Barcode</i> | 9 |
| 2.3.1 <i>Barcode</i> ITF-6 | 10 |
| 2.3.2 Struktur <i>Barcode</i> ITF-6 | 11 |
| 2.4 <i>Barcode Scanner</i> | 11 |
| 2.4.1 <i>Module MCR12</i> | 12 |
| 2.5 Motor DC..... | 13 |
| 2.6 Arduino | 13 |
| 2.6.1 <i>Hardware</i> Arduino | 14 |
| 2.6.2 <i>Software</i> Arduino..... | 15 |

| | |
|---|-----|
| 2.7 Komunikasi Serial | 16 |
| 2.7.1 MAX232 | 16 |
| 2.7.2 RS232..... | 17 |
| 2.9 LCD..... | 17 |
| BAB III PERANCANGAN <i>HARDWARE</i> DAN <i>SOFTWARE</i> | 19 |
| 3.1 Diagram Fungsional Alat | 19 |
| 3.2 Perancangan Perangkat Keras (<i>Hardware</i>) | 20 |
| 3.2.1 Perancangan Konveyor | 20 |
| 3.2.2 Perancangan <i>Barcode</i> ITF-6..... | 21 |
| 3.2.3 Perancangan <i>Panel Box</i> | 21 |
| 3.3 Perancangan Elektronik..... | 22 |
| 3.3.1 <i>Wiring</i> LCD 16 x 4 dan Arduino Mega 2560 R3..... | 23 |
| 3.3.2 <i>Wiring Relay</i> dan Arduino Mega 2560 R3..... | 24 |
| 3.3.3 <i>Wiring Power Supply</i> dan <i>Relay</i> | 25 |
| 3.3.4 <i>Wiring Relay</i> dan Motor | 26 |
| 3.3.5 <i>Wiring Power Supply</i> dan Motor | 28 |
| 3.3.6 <i>Wiring Barcode Scanner</i> | 29 |
| 3.3.7 Rangkaian <i>Voltage Regulator</i> | 30 |
| 3.4 Perancangan Perangkat Lunak (<i>Software</i>) | 30 |
| BAB IV PENGUJIAN DAN ANALISA | 35 |
| 4.1 Pengukuran Jarak Antara MCR12 dan <i>Barcode</i> | 35 |
| 4.2 Pengukuran Kecepatan MCR12 Me- <i>scan Barcode</i> | 37 |
| 4.3 Pengujian Posisi <i>Barcode</i> | 43 |
| 4.4 Tampilan LCD | 44 |
| 4.5 Pengujian Sistem Secara Keseluruhan | 46 |
| BAB V PENUTUP | 57 |
| 5.1 Kesimpulan | 57 |
| 5.2 Saran..... | 57 |
| DAFTAR PUSTAKA | 59 |
| LAMPIRAN A | A-1 |
| LAMPIRAN B | B-1 |
| LAMPIRAN C | C-1 |
| RIWAYAT HIDUP PENULIS | D-1 |

DAFTAR GAMBAR

| | HALAMAN |
|--|---------|
| Gambar 2.1 <i>Belt Conveyor</i> | 6 |
| Gambar 2.2 <i>Plastic Flat Chain Conveyor</i> | 6 |
| Gambar 2.3 <i>Vibrating Conveyor</i> | 7 |
| Gambar 2.4 <i>Hanging atau Overhead Conveyor</i> | 7 |
| Gambar 2.5 <i>Modul Relay 8 Channel</i> | 8 |
| Gambar 2.6 <i>Rangkaian Solid State Relay (SSR)</i> | 9 |
| Gambar 2.7 <i>Contoh Barcode ITF-6</i> | 10 |
| Gambar 2.8 <i>Ilustrasi Barcode Scanner</i> | 11 |
| Gambar 2.9 <i>Module MCR12</i> | 12 |
| Gambar 2.10 <i>Decoder Data Output Connector</i> | 12 |
| Gambar 2.11 <i>Motor DC</i> | 13 |
| Gambar 2.12 <i>Arduino Mega 2560 R3</i> | 14 |
| Gambar 2.13 <i>Simbol Software IDE</i> | 15 |
| Gambar 2.14 <i>Konfigurasi Pin MAX232</i> | 16 |
| Gambar 2.15 <i>Konfigurasi Pin RS232 (9 Pin Female)</i> | 17 |
| Gambar 2.16 <i>LCD 16 x 4</i> | 18 |
| Gambar 3.1 <i>Diagram Fungsional Alat</i> | 19 |
| Gambar 3.2 <i>Rancangan Konveyor</i> | 20 |
| Gambar 3.3 <i>Rancangan Barcode ITF-6</i> | 21 |
| Gambar 3.4 <i>Panel Box Tampak Dalam</i> | 22 |
| Gambar 3.5 <i>Panel Box Tampak Luar</i> | 22 |
| Gambar 3.6 <i>Wiring LCD 16 x 4 dan Arduino Mega 2560 R3</i> | 23 |
| Gambar 3.7 <i>Wiring Relay dan Arduino Mega 2560 R3</i> | 24 |
| Gambar 3.8 <i>Wiring Power Supply dan Relay</i> | 25 |
| Gambar 3.9 <i>Wiring Relay dan Motor</i> | 26 |
| Gambar 3.10 <i>Rangkaian Relay dan Lift Motor</i> | 27 |
| Gambar 3.11 <i>Rangkaian Relay dan Pusher Motor</i> | 27 |
| Gambar 3.12 <i>Wiring Power Supply dan Motor</i> | 28 |
| Gambar 3.13 <i>Wiring Max232, RS232, DB 9 Female</i> | 29 |
| Gambar 3.14 <i>Rangkaian Voltage Regulator</i> | 30 |
| Gambar 3.15 <i>Flowchart</i> | 31 |
| Gambar 3.16 <i>Program Inisialisasi Alamat pada LCD</i> | 32 |
| Gambar 3.17 <i>Program Inisialisasi Scanner Barcode</i> | 32 |
| Gambar 3.18 <i>Program Inisialisasi Relay</i> | 33 |
| Gambar 4.1 <i>Pengukuran Jarak MCR12 Me-scan Barcode</i> | 36 |

| | | |
|-------------|---|----|
| Gambar 4.2 | Ketika Me-scan <i>Barcode</i> (5cm) | 38 |
| Gambar 4.3 | Ketika Me-scan <i>Barcode</i> (10cm) | 39 |
| Gambar 4.4 | Ketika Me-scan <i>Barcode</i> (15cm) | 40 |
| Gambar 4.5 | Ketika Me-scan <i>Barcode</i> (20cm) | 41 |
| Gambar 4.6 | Ketika Me-scan <i>Barcode</i> (25cm) | 42 |
| Gambar 4.7 | Hasil Kecepatan Waktu MCR12 Me-scan <i>Barcode</i> | 43 |
| Gambar 4.8 | Pengujian Posisi Ketika Mendeteksi <i>Barcode</i> | 44 |
| Gambar 4.9 | LCD Sebelum <i>Scanner</i> Mendeteksi <i>Barcode</i> | 45 |
| Gambar 4.10 | LCD Setelah <i>Scanner</i> Mendeteksi <i>Barcode</i> | 45 |
| Gambar 4.11 | LCD Setelah Konveyor Mengelompokkan Barang | 46 |
| Gambar 4.12 | LCD Setelah <i>Scanner</i> Mendeteksi Beberapa <i>Barcode</i> ... | 46 |
| Gambar 4.13 | <i>Scanner Barcode</i> Mendeteksi <i>Barcode A</i> | 47 |
| Gambar 4.14 | Barang Melewati Konveyor 2 | 48 |
| Gambar 4.15 | Barang Melewati Konveyor 3 | 48 |
| Gambar 4.16 | Barang Sampai di Ujung Konveyor | 48 |
| Gambar 4.17 | Kondisi Ketika <i>Barcode A</i> Dikelompokkan | 49 |
| Gambar 4.18 | LCD Setelah Konveyor Mengelompokkan Barang A.... | 49 |
| Gambar 4.19 | <i>Scanner Barcode</i> Mendeteksi <i>Barcode B</i> | 51 |
| Gambar 4.20 | Barang Naik <i>Lift</i> | 51 |
| Gambar 4.21 | Barang Melewati Konveyor 4 | 51 |
| Gambar 4.22 | Kondisi Ketika <i>Barcode B</i> Dikelompokkan..... | 51 |
| Gambar 4.23 | LCD Setelah Konveyor Mengelompokkan Barang B | 52 |
| Gambar 4.24 | <i>Scanner Barcode</i> Mendeteksi <i>Barcode C</i> | 53 |
| Gambar 4.25 | <i>Pusher</i> Mendorong Barang | 54 |
| Gambar 4.26 | Kondisi Ketika <i>Barcode C</i> Dikelompokkan | 54 |
| Gambar 4.27 | LCD Setelah Konveyor Mengelompokkan Barang C ... | 54 |

DAFTAR TABEL

| | HALAMAN |
|---|---------|
| Tabel 2.1 Konfigurasi Konektor MCR12 | 12 |
| Tabel 2.2 Spesifikasi Arduino Mega 2560 R3 | 14 |
| Tabel 2.3 Spesifikasi LCD 16 x 4 | 18 |
| Tabel 3.1 <i>Wiring</i> Pin LCD 16x4 dan Arduino Mega 2560 R3 | 23 |
| Tabel 3.2 <i>Wiring</i> Pin <i>Relay</i> dan Arduino Mega 2560 R3 | 24 |
| Tabel 3.3 <i>Wiring Power Supply</i> dan <i>Relay</i> | 25 |
| Tabel 3.4 <i>Wiring Motor</i> dan <i>Relay</i> | 26 |
| Tabel 3.5 <i>Wiring Power Supply</i> dan Motor | 28 |
| Tabel 3.6 <i>Wiring</i> Komunikasi Serial dan Arduino Mega 2560 | 29 |
| Tabel 4.1 Jumlah <i>Barcode</i> Ter- <i>scan</i> | 36 |
| Tabel 4.2 Jumlah Rata-Rata <i>Barcode</i> Ter- <i>scan</i> | 37 |
| Tabel 4.3 Hasil Kecepatan MCR12 Me- <i>scan Barcode</i> (5cm) | 38 |
| Tabel 4.4 Hasil Kecepatan MCR12 Me- <i>scan Barcode</i> (10cm) | 39 |
| Tabel 4.5 Hasil Kecepatan MCR12 Me- <i>scan Barcode</i> (15cm) | 40 |
| Tabel 4.6 Hasil Kecepatan MCR12 Me- <i>scan Barcode</i> (20cm) | 41 |
| Tabel 4.7 Hasil Kecepatan MCR12 Me- <i>scan Barcode</i> | 42 |
| Tabel 4.8 Hasil Pengujian Posisi <i>Barcode</i> ITF-6 | 43 |
| Tabel 4.9 Hasil Pengujian <i>Barcode</i> A | 49 |
| Tabel 4.10 Hasil Pengujian <i>Barcode</i> B | 52 |
| Tabel 4.11 Hasil Pengujian <i>Barcode</i> C | 55 |

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di perindustrian saat ini, perusahaan selalu berupaya memenuhi kebutuhan masyarakat secara maksimal dengan mengganti pekerjaan yang selama ini dilakukan manusia untuk digantikan dengan mesin-mesin dalam rangka peningkatan kualitas produksinya. Seperti contoh, proses produksi atau proses *packaging* yang pada awalnya masih dilakukan secara manual digantikan dengan mesin. Pada industri manual tersebut, terdapat beberapa kekurangan. Salah satunya, waktu yang dibutuhkan dalam pengerjaan relatif lebih lama. Sehingga, dalam proses distribusi dibutuhkan suatu alat untuk mendistribusikan barang tersebut dengan waktu yang relatif lebih cepat, yaitu, dengan menggunakan konveyor. Dengan konveyor, manusia hanya perlu meletakkan barang di titik-titik konveyor yang telah ditentukan, kemudian didistribusikan ke tempat lainnya sehingga waktu yang dibutuhkan akan lebih cepat.

Di toko kebutuhan, hampir seluruh barang-barang menggunakan *barcode* di setiap kemasannya yang digunakan sebagai sarana untuk mempermudah kita dalam memasukkan data secara otomatis ketika membaca kode. *Barcode* adalah kode batang dengan garis hitam diatas *background* putih. Jenis *scanner barcode* yang banyak digunakan adalah tipe ITF-6. Di industri, adanya *barcode* dapat memudahkan jalannya proses distribusi karena kita dapat mengelompokkan barang berdasarkan *barcode-barcode* tersebut. Dengan adanya konveyor dan *scanner barcode* maka jalannya proses distribusi dapat lebih maksimal lagi.

Oleh karena itu, perusahaan dapat melakukan inovasi dari alat-alat tersebut. Dan konveyor akan membantu jalannya proses pengelompokan tersebut. Di beberapa industri, masih ada yang menggunakan cara manual untuk mengelompokkan barang-barang. Seperti contoh, barang A dan B dikelompokkan ke tempatnya masing-masing yang dilakukan oleh seorang pekerja. Kekurangan industri manual lainnya tersebut, yaitu terjadinya *human error* ketika proses pengerjaan. Pekerja tersebut dapat melakukan kesalahan, seperti, salah memasukkan barang ke tempat yang sesuai karena durasi waktu bekerja yang lama. Sehingga, untuk mengatasi masalah

tersebut, industri yang menginginkan proses produksi yang lebih efektif dan efisien, dapat melakukan perubahan pola produksi dengan mengaplikasikan sistem otomatisasi dalam produksinya. Seperti halnya, dalam mengelompokkan barang berdasarkan *barcode* produk akan membutuhkan sebuah alat yang dapat mengelompokkan secara otomatis.

Di industri, laporan untuk jalannya setiap proses juga tidak kalah penting. Sehingga, dibutuhkan alat yang dapat memantau mesin-mesin yang telah dikembangkan tersebut melalui sebuah layar sehingga akan mempermudah bagi pekerja industri untuk melakukan pengecekan, baik pengecekan setiap barang yang didistribusikan, maupun pengecekan adanya kerusakan pada alat. Untuk memantau alat-alat tersebut dapat menggunakan LCD sesuai kebutuhan. LCD akan menampilkan hasil perhitungan dan pengelompokan yang terdeteksi dari *scanner barcode*. Program untuk menghubungkan komunikasi antara LCD, konveyor, dan *scanner barcode* dapat menggunakan Arduino Mega.

Prototype konveyor ini dikembangkan dari alat yang sudah ada sebelumnya, yaitu milik Andrik Kurnia Pratama dan Rahmat Bagus Prasjo di tahun 2015, dengan judul Tugas Akhir “Mesin Pengelompok Barang Berdasarkan Ketinggian dan Konveyor Pengangkut Menggunakan PLC”. Begitu pula dengan *scanner barcode* MCR12 yang dikembangkan lagi dari judul sebelumnya, yaitu milik Fadila Lingga Dewi di tahun 2017, dengan judul “Alat Penghitung Jumlah Barang Menggunakan *Barcode* ITF-14”.

1.2 Permasalahan

Dalam sistem distribusi di perindustrian, sering terjadi kesalahan ketika memasukkan data akibat *human error*. Dan, proses distribusi di industri manual dianggap masih kurang maksimal karena waktu yang kurang efektif.

1.3 Batasan Masalah

Pada Tugas Akhir ini, batasan masalah yang diambil adalah :

1. *Scanner barcode* MCR12 mendeteksi *barcode* ITF-6 dengan jarak 15cm.
2. Posisi *barcode* dari setiap barang diletakkan dengan posisi 90° atau 270° dalam rotasi horizontal sesuai kedudukan *barcode scanner*.

3. Contoh *barcode* yang digunakan berjumlah tiga(3) barang dengan tipe *barcode* ITF-6.

1.4 Tujuan

Tujuan yang akan dicapai dari Tugas Akhir ini adalah sebagai berikut:

1. Merancang deteksi *barcode* pada barang yang melewati konveyor.
2. Membuat suatu sistem otomatis yang dapat melakukan perhitungan dan pengelompokan barang berlabel *barcode* berdasarkan jenisnya dengan tepat.
3. Membuat suatu sistem untuk *monitoring* variabel jumlah dan jenis barang yang ditampilkan di LCD.
4. Mengimplementasikan *hardware* dan *software* yang tepat di konveyor.

1.5 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan dan relevansi.

Bab II Teori Penunjang

Bab ini menjelaskan tentang tinjauan pustaka dari konveyor, *relay*, motor DC, Arduino Mega 2560, *barcode scanner*, *barcode*, *software* Arduino IDE, dan LCD.

Bab III Perancangan Hardware dan Software

Bab ini membahas perencanaan dan pembuatan perangkat keras (*Hardware*) yang meliputi desain alat serta pengimplementasian sensor yang digunakan, untuk perhitungan dan pengelompokan barang, serta pembuatan perangkat lunak (*Software*) yang meliputi program pada Arduino IDE untuk menjalankan konveyor.

Bab IV Pengujian dan Analisis

Bab ini memuat tentang pemaparan dan analisis hasil pengujian alat pada keadaan sebenarnya. Seperti pengujian *barcode scanner*, serta pengujian keseluruhan alat terhadap barang ketika di konveyor.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.6 Relevansi

Diharapkan alat *monitoring* berdasarkan *barcode scanner* di konveyor ini dapat mengurangi kesalahan karena adanya *human error*, serta dapat dijadikan sebagai bahan pembelajaran mengenai proses distribusi di perindustrian.

BAB II

TEORI PENUNJANG

Pada bab ini membahas mengenai teori-teori dari peralatan yang digunakan dalam Tugas Akhir yang berjudul *Monitoring Perhitungan dan Pengelompokan Barang di Konveyor Berdasarkan Kode Batang (Scanner Barcode) Melalui LCD*. Teori yang mendukung penyelesaian Tugas Akhir ini diantaranya adalah mengenai Konveyor, *Relay*, Motor DC, Arduino Mega 2560 R3, *Software* Arduino IDE, *Barcode*, *Scanner Barcode*, Komunikasi Serial, dan LCD. Proyek tugas akhir ini memiliki perbedaan dengan tugas akhir lainnya yang menggunakan konveyor dan *scanner barcode* yang sama. Dimana terdapat perbedaan kontrol pada konveyor dan program deteksi *barcode* yang digunakan. Pada tugas akhir ini menggunakan kontrol Arduino dengan menggunakan Arduino Mega 2560 R3 dan program untuk deteksi *barcode* yang digunakan yaitu untuk mengelompokkan dan menghitung barang yang terdeteksi yang dipasang di konveyor berdasarkan *scanner barcode* MCR12.

2.1 Konveyor [1]

Konveyor adalah suatu sistem mekanik yang mempunyai fungsi memindahkan barang dari satu tempat ke tempat yang lain. Konveyor banyak dipakai di industri untuk transportasi barang yang jumlahnya sangat banyak dan berkelanjutan. Konveyor berguna dalam aplikasi yang melibatkan transportasi bahan berat atau besar. Banyak jenis sistem konveyor yang tersedia, dan digunakan sesuai dengan kebutuhan berbagai industri yang berbeda. Berikut adalah jenis-jenis konveyor.

2.1.1 Belt Conveyor

Alat ini memiliki penggunaan yang sangat luas dalam berbagai industri. Di area industri pertambangan, faktor ukuran (ketebalan dan lebar), dan ketahanan (*reability*) menjadi faktor yang utama. Penggunaan di operasi manufaktur juga tentu berbeda, meski memiliki spesifikasi yang lebih rendah, alat ini tetap merupakan bagian yang sangat penting dalam rangkaian sistem operasi. Bentuk *belt conveyor* secara umum dapat dilihat pada Gambar 2.1.



Gambar 2.1 *Belt Conveyor*

2.1.2 *Plastic Flat Chain Conveyor*

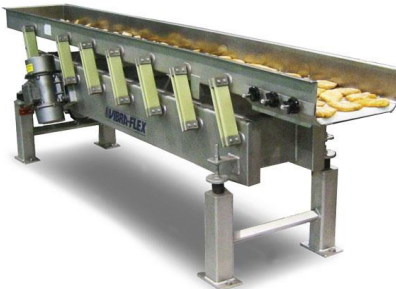
Konveyor ini biasa ditemukan di industri makanan dan minuman kemasan. Prinsipnya, sama dengan cara kerja rantai *tank* tempur. Keunggulan sistem ini yaitu permukaan pada bagian atas konveyor rata atau *flat*, dan dapat bergerak melengkung sehingga efektif digunakan ketika area kerja memiliki jarak yang sedikit, namun proses kerja memerlukan konveyor dengan jarak yang cukup panjang untuk mengantisipasi terjadinya antrian produk. Bentuk *plastic flat chain conveyor* dapat dilihat pada Gambar 2.2.



Gambar 2.2 *Plastic Flat Chain Conveyor*

2.1.3 *Vibrating Conveyor*

Alat ini bekerja menggunakan efek getaran atau vibrasi mekanis yang menggerakkan barang secara vertikal atau horizontal. Bentuk dari *vibrating conveyor* dapat dilihat pada Gambar 2.3.



Gambar 2.3 *Vibrating Conveyor*

2.1.4 *Hanging atau Overhead Conveyor*

Alat ini berfungsi untuk memindahkan barang dengan cara digantung. Untuk tenaga penggeraknya dapat menggunakan gravitasi atau motor. Bentuk *overhead conveyor* ini dapat dilihat pada Gambar 2.4.



Gambar 2.4 *Hanging atau Overhead Conveyor*

2.2 *Relay [2]*

Relay adalah saklar (*switch*) yang dioperasikan secara listrik dan merupakan komponen elektromekanikal (*electromechanical*) yang terdiri dari 2 bagian utama yaitu elektromagnet berupa induktor inti besi (*coil*) dan mekanikal (seperangkat kontak saklar atau *switch*). Saklar pada *relay* akan mengalami perubahan posisi *OFF* ke *ON* ketika diberikan energi elektromagnetik pada *armature relay* tersebut.. *Relay* yang ada di pasaran terdapat berbagai bentuk, ukuran tegangan kerja dan jumlah saklar yang bervariasi.

Relay yang digunakan untuk Tugas Akhir ini yaitu *Module Relay*, dimana *Module Relay* ini dapat digunakan sebagai *switch* untuk menjalankan berbagai peralatan elektronik. Misalnya lampu listrik, motor listrik, dan berbagai peralatan elektronik lainnya. kendali *ON / OFF switch (relay)*, sepenuhnya ditentukan oleh nilai *output* sensor, yang setelah diproses mikrokontroler akan menghasilkan perintah kepada *relay* untuk melakukan fungsi *ON / OFF*.

Relay merupakan salah satu jenis dari saklar, maka istilah *pole* dan *throw* yang dipakai dalam saklar juga berlaku pada *relay*. Berdasarkan penggolongan jumlah *pole* dan *throw*-nya, sebuah *module relay* tersebut dapat digolongkan dalam golongan *relay* SPDT karena memiliki 1 *Common*, 1 *Normally Open* (NO), dan 1 *Normally Close* (NC). Berikut pada Gambar 2.5 adalah gambar *Module Relay 8 Channel*.



Gambar 2.5 *Module Relay 8 Channel*

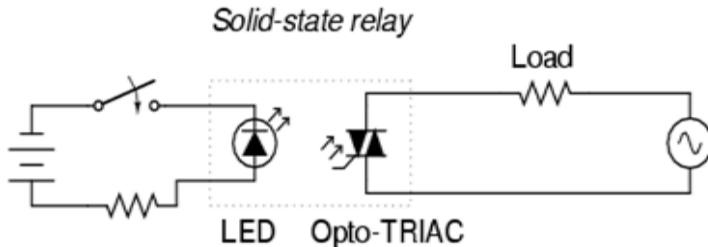
Di dalam *module relay* ini terdapat sebuah *relay* elektronik yang disebut SSR (*Solid State Relay*). Yaitu, *relay* yang tidak menggunakan kontaktor mekanik. *Solid state relay* menggunakan kontaktor berupa komponen aktif seperti TRIAC, sehingga *solid state relay* dapat dikendalikan dengan tegangan rendah dan dapat digunakan untuk mengendalikan tegangan AC dengan voltase besar. Pada sebagian besar aplikasi, SSR digunakan sebagai perantara antara rangkaian control tegangan rendah dengan tegangan *line* arus AC yang lebih tinggi. *Solid State Relay* ini memiliki keuntungan yaitu :

- a. Lebih terpercaya.
- b. Mempunyai umur pemakaian yang panjang karena SSR tidak mempunyai bagian yang bergerak dan digabungkan dengan transistor dan sirkuit IC.

- c. Tahan terhadap guncangan dan getaran.
- d. Mempunyai waktu respon yang lebih cepat.

Rangkaian *Solid State Relay* (SSR) dapat dilihat pada Gambar

2.6.



Gambar 2.6 Rangkaian *Solid State Relay* (SSR)

2.3 *Barcode* [3]

Barcode dapat diartikan sebagai kumpulan kode yang berbentuk garis, dimana masing-masing ketebalan setiap garis berbeda sesuai dengan isi kodenya. *Barcode* adalah informasi terbaca mesin (*machine readable*) dalam format visual yang tercetak. *Barcode* dibaca dengan menggunakan sebuah alat baca *barcode* atau lebih dikenal dengan *barcode scanner*. Merk *barcode scanner* yang terkenal diantaranya DATALOGIC, PSC, HHP, CHIPERLAB, ZEBEX, dan lain-lain. Seiring semakin bertambahnya penggunaan *barcode*, kini *barcode* tidak hanya bisa mewakili karakter angka saja tapi sudah meliputi seluruh kode ASCII. Manfaat penggunaan *barcode* antara lain :

1. Meningkatkan akurasi dengan mengurangi risiko kesalahan yang dilakukan manusia dari proses pemasukan data secara manual.
2. Dapat memaksimalkan proses pengumpulan data secara otomatis.
3. Data dapat diterima dengan cepat.
4. Meningkatkan efisiensi operasional sehingga memungkinkan sebuah perusahaan dapat mengemat biaya dan dapat meningkatkan keuntungan bisnis yang dijalani.

Di pasaran saat ini banyak jenis-jenis *barcode* yang sering digunakan. Ada beberapa standarisasi jenis *barcode*. Berikut ini adalah jenis *barcode* yang sering digunakan :

1. *Code 39* : sebagai simbolik yang paling populer di dunia barcode non-retail, dengan variabel digit yang panjang. Namun saat ini *code 39* makin sedikit dipergunakan dan digantikan dengan *Code 128* yang lebih mudah dibaca oleh pemindai.
2. *Code 128* : digunakan dalam preferensi untuk *Code 39* karena lebih kompak.
3. *Interleaved 2 of 5* : digunakan dalam industri pelayaran dan gudang.
4. *PDF417* : suatu jenis *barcode 2-D* baru yang dapat *encode* sampai 1108 *byte* informasi dan dapat terkompresi seperti pada sebuah *Portable Data File* (PDF).

Dari beberapa *barcode* yang sering digunakan tersebut, berikut adalah tipe *barcode* yang digunakan pada alat ini.

2.3.1 *Barcode ITF-6* [3]

ITF-6 (*Interleaved Two of Five*) adalah versi singkat dari ITF 14, yaitu implementasi GS1 dari *Interleaved 2 of 5 barcode* untuk mengkodekan *Item Number* Perdagangan Global. ITF-6 akan selalu mengkodekan 6 digit. Satu-satunya faktor pembatas untuk panjang kode ITF adalah kemampuan alat baca yang akan digunakan untuk membaca kode tersebut dan juga bahwa ITF harus memiliki jumlah digit genap. ITF digunakan untuk aplikasi industri dimana kode angka saja sudah mencukupi dan juga digunakan dalam lingkungan penjualan eceran untuk menandai bungkus luar atau packing bagian luar. Contoh *Barcode* ITF-6 ditunjukkan pada Gambar 2.7.



Gambar 2.7 Contoh *Barcode* ITF-6

2.3.2 Struktur *Barcode* ITF-6 [3]

Struktur dari *Barcode Interleaved 2 of 5* sebagai berikut:

1. *Indicator* : Untuk menunjukkan tingkat kemasan untuk karton tertentu. Awalan satu digit ini dapat berkisar antara 0 hingga 9.
2. *GS1 Company Prefix* : Tergantung pada jumlah *item* perusahaan yang perlu untuk diidentifikasi, sebuah GS1 Awalan Perusahaan mungkin antara 7 sampai 10 digit.
3. *Item Reference* : Referensi nomor produk yang sama digunakan untuk GTIN tingkat *item* (GTIN-8, GTIN-12, GTIN-13) ketika karton terdiri dari *item* yang sama. Untuk karton yang berisi bermacam-macam barang sejumlah produk baru yang ditugaskan.
4. *Check Digit* : *Digit* terakhir dari GTIN-14 adalah cek *digit* dihitung. Menggunakan algoritma MOD 10 cek *digit*, cek *digit* dihitung mencegah kesalahan substitusi.

2.4 *Barcode Scanner* [3]

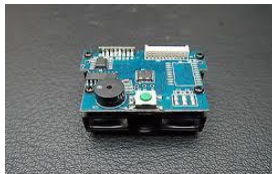
Barcode Scanner adalah piranti yang berfungsi membaca data dalam bentuk *barcode*. *Barcode* biasanya ditemukan melekat pada produk-produk *barcode* yang berupa deretan baris tegak dengan ketebalan yang bermacam-macam. Fungsi *barcode scanner* adalah untuk membaca sebuah kode yang berbentuk kotak-kotak maupun berbentuk garis tebal *vertikal* yang kemudian diterjemahkan dalam bentuk angka-angka. Cara kerja dari *barcode scanner* tersebut yaitu memindai simbol, menangkap dan merubah kode bar menjadi data elektrik lalu mengirimkannya ke komputer. Ilustrasi saat *barcode* di-scan oleh *barcode scanner* dapat dilihat pada Gambar 2.8.



Gambar 2.8 Ilustrasi *Barcode Scanner*

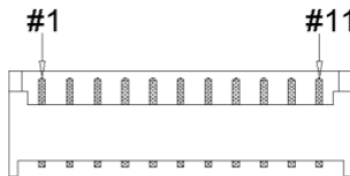
2.4.1 Module MCR12 [3]

Module MCR12 adalah *module* pemindai kode batang CCD (*Charge Couple Device*) jarak jauh dengan sensor gambar liner sensitif tinggi dan dibangun dalam fungsi *auto-sense*. Sebagai kemampuan operasi jarak jauh, ketahanan cahayanya sekitar 5.000 LUX, dan 100 pemindaian per detik. MCR12 *decode board* didukung oleh prosesor yang cepat dan dapat memecahkan beragam kode batang 1D. Bentuk fisik dari MCR12 dapat dilihat pada Gambar 2.9.



Gambar 2.9 Module MCR12

Untuk konfigurasi dari MCR12 dapat dilihat pada Gambar 2.10 dan tabel spesifikasinya pada Tabel 2.1.



Gambar 2.10 Decoder Data Output Connector

Tabel 2.1 Konfigurasi Konektor MCR12

| Pin | Keterangan |
|-----|------------------|
| 1 | GND |
| 2 | VCC (+5V) |
| 3 | TXD |
| 4 | RXD |
| 5 | <i>Host Data</i> |
| 6 | <i>Host CLK</i> |
| 7 | KB Data |

| Pin | Keterangan |
|-----|---------------|
| 8 | KB CLK |
| 9 | RTS |
| 10 | CTS |
| 11 | <i>Shield</i> |

2.5 Motor DC [4]

Motor listrik merupakan perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor DC disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet, maka akan timbul tegangan (GGL) yang berubah-ubah arah pada setiap setengah putaran, sehingga merupakan tegangan bolak-balik.

Prinsip kerja dari motor DC adalah setiap konduktor yang mengalirkan arus memiliki medan magnet disekelilingnya. Kuat medan tergantung pada besarnya arus yang mengalir pada konduktor. Motor DC yang digunakan dapat dilihat pada Gambar 2.11.



Gambar 2.11 Motor DC

2.6 Arduino [5]

Arduino merupakan perangkat elektronik atau papan rangkaian elektronik *open – source* yang di dalamnya terdapat komponen utama, yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan Atmel. Mikrokontroler itu sendiri adalah *chip*

atau IC (*Integrated circuit*) yang bisa diprogram menggunakan komputer. Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca *input*, memproses *input* tersebut dan kemudian menghasilkan *output* sesuai yang diinginkan. Sehingga mikrokontroler bertugas sebagai otak yang mengendalikan *input*, proses, dan *output* sebuah rangkaian elektronik.

2.6.1 *Hardware* Arduino

Perangkat keras dalam arduino memiliki beberapa jenis, yang mempunyai kelebihan dan kekurangan dalam setiap papannya. Penggunaan jenis arduino disesuaikan dengan kebutuhan, hal ini yang akan mempengaruhi dari jenis prosesor yang digunakan. Jika semakin kompleks perancangan dan program yang dibuat, maka harus sesuai pula jenis *controller* yang digunakan. Yang membedakan antara arduino satu dengan yang lainnya adalah penambahan fungsi dalam setiap papan sirkuitnya serta jenis mikrokontroler yang digunakan. Dalam Tugas Akhir ini, jenis arduino yang digunakan adalah Arduino Mega 2560 R3 yang ditunjukkan pada Gambar 2.12.



Gambar 2.12 Arduino Mega 2560 R3

Tabel 2.2 Spesifikasi Arduino Mega 2560 R3

| Spesifikasi | Keterangan |
|-------------------------------------|------------|
| Tegangan Operasi | 5 Volt |
| Spesifikasi | Keterangan |
| Tegangan <i>Input</i> (Rekomendasi) | 7-12 Volt |
| Tegangan <i>Input</i> (Limit) | 6-20 Volt |

| Spesifikasi | Keterangan |
|----------------------------|---|
| Pin <i>Digital I/O</i> | 54 Pin (15 Pin dapat sebagai PWM) |
| Pin <i>Analog Input</i> | 16 Pin |
| Arus DC Per Pin <i>I/O</i> | 20 mA |
| Arus DC untuk Pin 3.3 Volt | 50 mA |
| <i>Flash Memory</i> | 258 KB (8 KB digunakan oleh <i>bootloader</i>) |
| <i>SRAM</i> | 8 KB |
| Spesifikasi | Keterangan |
| <i>EEPROM</i> | 4 KB |
| <i>Clock Speed</i> | 16 MHz |
| Pin <i>Digital I/O</i> | 54 (15 dapat sebagai PWM) |
| <i>LED_BUILTIN</i> | 13 |
| <i>Panjang</i> | 101.52 mm |
| <i>Luas</i> | 50 mA |
| <i>Berat</i> | 37 gram |

2.3.3 *Software Arduino*

Arduino IDE adalah perangkat lunak yang digunakan untuk memrogram, *monitoring* dan *debugging* mikrokontroler Arduino. IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Berikut adalah simbol dari *software* IDE pada Gambar 2.13.



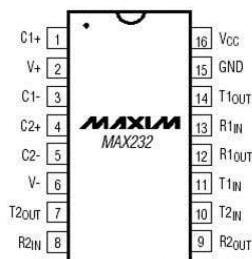
Gambar 2.13 Simbol *Software* IDE

2.7 Komunikasi Serial [6]

Komunikasi serial adalah sebuah komunikasi yang terjadi dengan mengirimkan data *per-bit* secara berurutan dan bergantian. Komunikasi ini hanya membutuhkan satu jalur atau kabel yang sedikit dibandingkan dengan komunikasi paralel, di mana hanya satu bit data yang dikirimkan melalui seuntai jalur atau kabel pada waktu tertentu secara berurutan. Manfaat paling dasar adanya komunikasi serial ialah memungkinkan terjadinya pertukaran informasi antara papan Arduino dan komputer. Hal ini memungkinkan untuk dapat memantau status pengerjaan yang sedang dilakukan langsung di komputer. Sebagai contoh, dapat melihat data dari sensor yang sedang diuji di komputer. Dengan demikian fokus hanyalah pada sensor-sensor tersebut yang sedang diuji. Komunikasi serial pada Arduino dapat dilakukan secara dua arah. Artinya, baik komputer yang kita gunakan maupun papan Arduino yang sedang kita program dapat saling berkomunikasi dan saling mengirim maupun menerima informasi.

2.7.1 MAX232 [7]

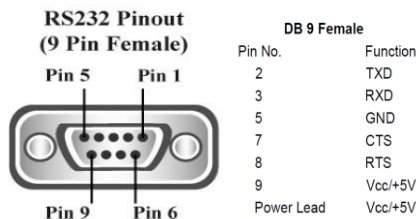
MAX232 merupakan salah satu jenis IC rangkaian antar muka dual RS-232 *transmitter/receiver* yang memenuhi semua spesifikasi standar EIA-232-E. IC MAX232 hanya membutuhkan *power supply* 5V (*single power supply*) sebagai catu. IC MAX232 berfungsi untuk merubah level tegangan pada COM1 menjadi level tegangan TTL / CMOS. IC MAX232 terdiri atas tiga bagian yaitu *dual charge-pump voltage converter*, *driver RS232*, dan *receiver RS232*. Konfigurasi dari MAX232 dapat dilihat pada Gambar 2.14.



Gambar 2.14 Konfigurasi Pin MAX232

2.7.2 RS232 [8]

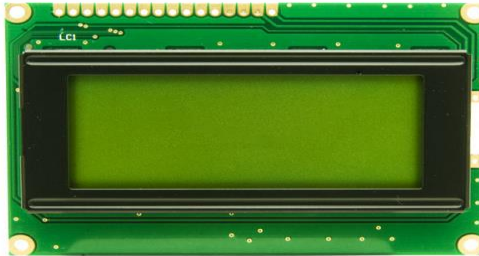
RS232 adalah standar komunikasi serial yang digunakan untuk koneksi *peripheral* ke *peripheral*. Biasa juga disebut dengan jalur I/O (*input/output*). Biasanya dihubungkan lewat jalur *port* serial RS232. Standar ini menggunakan beberapa piranti dalam implementasinya. Paling umum yang dipakai adalah *plug*/konektor DB9 atau DB25. Serial *port* RS232 pada konektor DB9 memiliki pin 9 buah dan pada konektor DB25 memiliki pin 25 buah. Fungsi dari *serial port* RS232 adalah untuk menghubungkan atau koneksi dari perangkat yang satu dengan perangkat yang lain, atau peralatan standar yang menyangkut komunikasi data antara komputer dengan alat-alat pelengkap komputer. Pin-pin dari RS232 dapat dilihat pada Gambar 2.15.



Gambar 2.15 Konfigurasi Pin RS232 (9 Pin *Female*)

2.9 LCD [9]

Liquid Crystal Display (LCD) adalah sebuah peralatan elektronik yang berfungsi untuk menampilkan *output* sebuah sistem dengan cara membentuk suatu citra atau gambaran pada sebuah layar. Secara garis besar, komponen penyusun LCD terdiri dari kristal cair (*liquid crystal*) yang diapit oleh 2 buah elektroda transparan dan 2 buah filter polarisasi (*polarizing filter*). LCD yang ada dipasaran dikategorikan menurut jumlah baris yang dapat digunakan pada LCD yaitu 1 baris, 2 baris ,dan 4 baris yang dapat digunakan hingga 80 karakter. Umumnya LCD yang digunakan adalah LCD dengan 1 *controller* yang memiliki 14 pin. LCD yang digunakan pada alat ini adalah LCD 16 x 4. Bentuk fisik dari LCD dapat dilihat pada Gambar 2.16.



Gambar 2.16 LCD 16 x 4

Tabel 2.3 Spesifikasi LCD 16 x 4

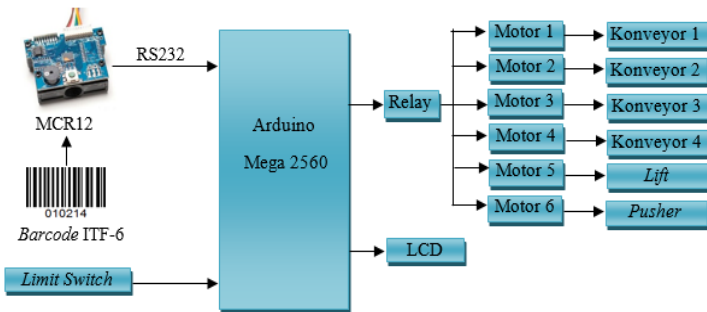
| Pin | Keterangan |
|-----|----------------------|
| 1 | VSS |
| 2 | VCC |
| 3 | VEE |
| 4 | RS |
| 5 | RW |
| 6 | EN |
| 7 | D0 |
| 8 | D1 |
| 9 | D2 |
| 10 | D3 |
| 11 | D4 |
| 12 | D5 |
| 13 | D6 |
| 14 | D7 |
| 15 | <i>Backlight +</i> |
| 16 | <i>Backlight GND</i> |

BAB III

PERANCANGAN *HARDWARE* DAN *SOFTWARE*

Perancangan Alat *Monitoring* Perhitungan dan Pengelompokan Barang di Konveyor Berdasarkan Kode Batang (*Scanner Barcode*) melalui LCD ini diawali dengan diagram fungsional alat yang menunjukkan konsep dan cara kerja alat pada TA ini. Kemudian dijelaskan tentang perancangan perangkat keras (*hardware*) meliputi perancangan *panel box*, perancangan konveyor, perancangan sensor, perancangan elektronik, dan *wiring* rangkaian *hardware*. Dan yang terakhir adalah pemaparan rancangan perangkat lunak (*software*) yang terdiri dari perancangan program Arduino. Sehingga proses monitoring sensor *barcode* ini diawali dengan pengolahan data dari sensor oleh Arduino kemudian diproses di konveyor dan data yang dihasilkan dapat ditampilkan ke LCD.

3.1 Diagram Fungsional Alat



Gambar 3.1 Diagram Fungsional Alat

Dari keseluruhan sistem pada Gambar 3.1 dapat dilihat bahwa ketika *barcode* ITF-6 yang ter-*scan* melalui *barcode scanner* MCR12 akan mengaktifkan *relay* yang telah diprogram melalui Arduino, sehingga *relay* akan menjalankan *driver motor* di konveyor sesuai program di Arduino. Motor yang digunakan pada alat ini berjumlah 6 buah. *Driver motor* akan mengaktifkan motor pada konveyor. Motor 1 menggerakkan konveyor 1, motor 2 menggerakkan konveyor 2, motor 3 menggerakkan konveyor 3,

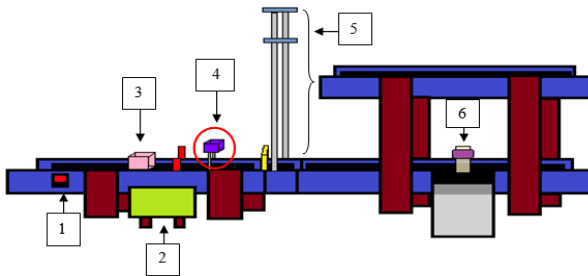
motor 4 menggerakkan konveyor 4, motor 5 menggerakkan *lift*, dan motor 6 untuk menggerakkan *pusher*. Ketika *barcode A* terdeteksi, maka konveyor 1, 2, 3 akan aktif dan barang akan berjalan lurus. Ketika *barcode B* terdeteksi, maka konveyor 1,2 akan bekerja kemudian *lift* akan naik dan ketika barang sudah sampai atas konveyor 4 akan aktif hingga barang jatuh dari konveyor atas. Setelah pengelompokan tersebut selesai, *lift* kembali turun ke keadaan semula. Dan jika *barcode C* yang terdeteksi, maka konveyor 1, 2, 3 akan bekerja, ketika barang sampai di *pusher* maka *pusher* akan mulai mendorong barang hingga jatuh. Setelah pengelompokan di konveyor, LCD akan menampilkan hasil dari deteksi *barcode* mengenai jumlah setiap *barcode* dan jumlah secara keseluruhan.

3.2 Perancangan Perangkat Keras (*Hardware*)

Perangkat keras yang dibuat dalam alat *monitoring* perhitungan dan pengelompokan barang di konveyor berdasarkan kode batang (*scanner barcode*) melalui LCD ini merupakan suatu simulator konveyor sebagai alat distribusi yang ada di perindustrian. Semua perangkat ini diintegrasikan sehingga dapat menjalankan suatu mekanisme kerja distribusi seperti di perindustrian.

3.2.1 Perancangan Konveyor

Konveyor adalah suatu sistem mekanik yang berfungsi memindahkan barang dari satu tempat ke tempat yang lain. Konveyor banyak dipakai di industri untuk transportasi barang maupun mendistribusikan barang yang jumlahnya sangat banyak dan berkelanjutan. Untuk Tugas Akhir ini menggunakan *belt conveyor*. Rancangan konveyor ditunjukkan oleh Gambar 3.2.



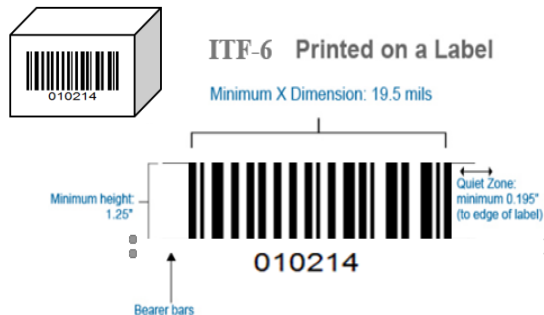
Gambar 3.2 Rancangan Konveyor

Keterangan Gambar 3.2 Rancangan Konveyor :

1. Tombol *ON/OFF* konveyor
2. LCD
3. Contoh barang (berlabel *barcode*)
4. *Scanner barcode* ITF-6
5. *Lift Conveyor*
6. *Pusher*

3.2.2 Perancangan *Barcode* ITF-6

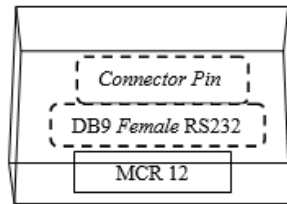
ITF-6 adalah *barcode* 6 digit yang menggunakan simbologi "*Interleaved 2 of 5*" (*I 2 of 5*, atau ITF). Simbol *barcode* ITF-6 mencakup balok pembawa, yang mengelilingi bar yang melindungi citra *barcode*. Jika ITF-6 dicetak langsung pada karton bergelombang maka harus dikelilingi oleh balok pembawa. Ilustrasi berikut menampilkan contoh gambar dan ukuran minimum untuk setiap skenario. Rancangan dari *barcode* ITF-6 ditunjukkan pada Gambar 3.3.



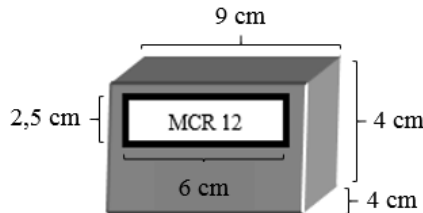
Gambar 3.3 Rancangan *Barcode* ITF-6

3.2.3 Perancangan *Panel Box*

Perancangan *panel box* dimaksudkan agar penempatan peralatan menjadi rapi dan mudah dimengerti sehingga dari segi estetika dan fungsinya menjadi lebih baik. Perancangan *panel box* tampak dalam dapat dilihat pada Gambar 3.4, sedangkan perancangan *panel box* tampak luar dapat dilihat pada Gambar 3.5.



Gambar 3.4 *Panel Box* Tampak Dalam



Gambar 3.5 *Panel Box* Tampak Luar

Keterangan Gambar 3.4 *Panel Box* Tampak Dalam dan Gambar 3.5 *Panel Box* Tampak Luar :

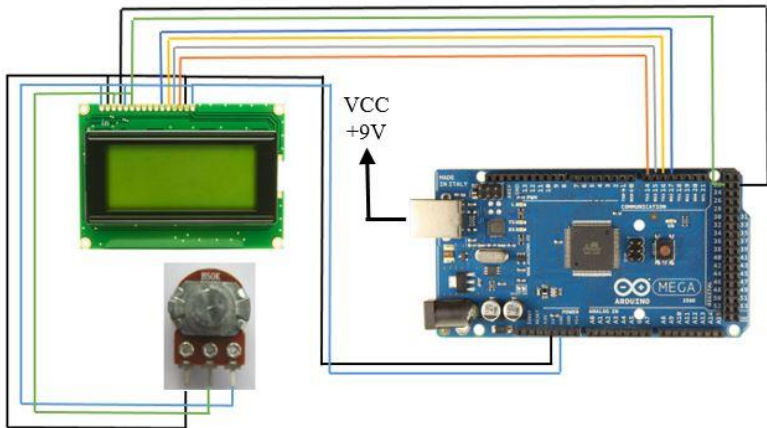
1. *Connector Pin*, berfungsi untuk menghubungkan satu rangkaian elektronika ke rangkaian elektronika lainnya ataupun untuk menghubungkan suatu perangkat dengan perangkat lainnya.
2. *DB9 Female RS232*, berfungsi untuk mengirimkan data dari sensor ke Arduino.
3. *MCR12*, adalah sensor kode batang atau *barcode scanner*.

3.3 Perancangan Elektronik

Pada Tugas Akhir ini terdapat beberapa perancangan elektronik yang akan digunakan agar alat dapat berjalan dengan baik. Berikut merupakan beberapa perancangan elektronik untuk alat *monitoring* perhitungan dan pengelompokan barang berdasarkan kode batang (*scanner barcode*) melalui LCD.

3.3.1 Wiring LCD 16 x 4 dan Arduino Mega 2560 R3

Display yang akan digunakan pada *project* Tugas Akhir ini adalah *display* LCD 16 x 4, artinya LCD terdiri dari 4 baris dan 16 karakter. Pada Gambar 3.6 adalah *wiring* LCD 16 x 4 dan Arduino Mega 2560 R3 yang juga menggunakan sebuah potensiometer dan VCC +9V. Untuk lebih jelasnya *wiring* antara pin LCD 16 x 4 dan pin pada Arduino Mega 2560 dapat dilihat pada Tabel 3.1.



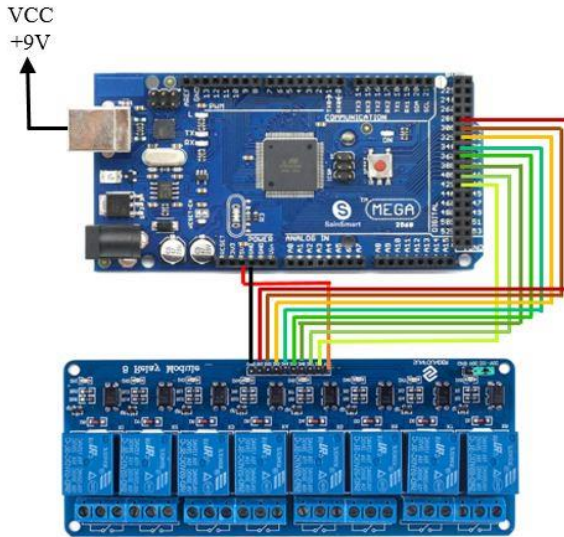
Gambar 3.6 Wiring LCD 16 x 4 dan Arduino Mega 2560 R3

Tabel 3.1 Wiring Pin LCD 16x4 dan Arduino Mega 2560 R3

| No. | Potensiometer | Pin LCD 16 x 4 | Pin Arduino Mega 2560 |
|-----|---------------|---------------------------------|-----------------------|
| 1. | Kaki 1 | <i>Blacklight</i> +, VCC | +5V |
| 2. | Kaki 2 | VEE, R/W | |
| 3. | Kaki 3 | <i>Blacklight</i> GND, R/W, VSS | |
| 4. | | D7 | Pin 14 |
| 5. | | D6 | Pin 15 |
| 6. | | D5 | Pin 16 |
| 7. | | D4 | Pin 17 |
| 8. | | EN | Pin 22 |
| 9. | | RS | Pin 23 |

3.3.2 Wiring Relay dan Arduino Mega 2560 R3

Relay yang digunakan untuk alat ini adalah *module relay 8 channel*. Relay digunakan untuk menyuplai *motor driver* pada konveyor dengan *input* +5V dan *output* +12V. Arduino di *supply* dengan VCC +9V. Untuk lebih jelasnya, *wiring* pin Relay dan Arduino Mega dapat dilihat pada Gambar 3.7 dan Tabel 3.2.



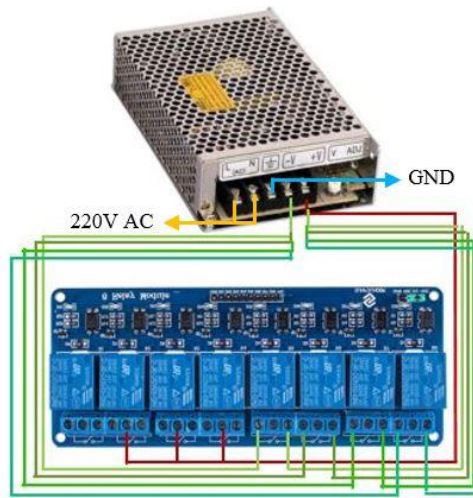
Gambar 3.7 Wiring Relay dan Arduino Mega2560 R3

Tabel 3.2 Wiring Pin Relay dan Arduino Mega 2560 R3

| No. | Pin Relay | Pin Arduino Mega 2560 |
|-----|-----------|-----------------------|
| 1 | GND | GND |
| 2 | VCC | +5V |
| 3 | IN1 | 28 |
| 4 | IN2 | 30 |
| 5 | IN3 | 32 |
| 6 | IN4 | 34 |
| 7 | IN5 | 36 |
| 8 | IN6 | 38 |
| 9 | IN7 | 40 |
| 10 | IN8 | 42 |

3.3.3 Wiring Power Supply dan Relay

Relay yang digunakan untuk alat ini adalah *relay module 8 channel* dan *power supply* yang digunakan adalah *power supply 12V 5A*. Untuk mengaktifkan motor, maka motor harus disambungkan ke *relay* dan *relay* harus mendapat *supply* sesuai kebutuhan motor. *Power supply* diberi supply 220V AC. Gambar dari *wiring power supply* dan *relay* beserta tabel keterangannya dapat dilihat pada Gambar 3.8 dan Tabel 3.3.



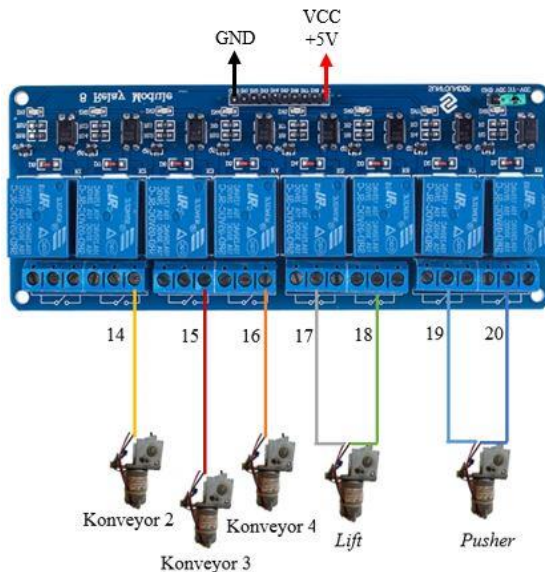
Gambar 3.8 Wiring Power Supply dan Relay

Tabel 3.3 Wiring Power Supply dan Relay

| No. | Power Supply | NO | NC | COM |
|-----|--------------|---------|---------|-------------|
| 1. | V+ | Relay 5 | | |
| 2. | | Relay 6 | | |
| 3. | | Relay 7 | | |
| 4. | | Relay 8 | | |
| 5. | | | | Relay 2,3,4 |
| 6. | V- | | Relay 5 | |
| 7. | | | Relay 6 | |
| 8. | | | Relay 7 | |
| 9. | | | Relay 8 | |

3.3.4 Wiring Relay dan Motor

Setelah *power supply* disambungkan *relay*, maka motor disambungkan *relay* agar ketika program dari Arduino berjalan dapat mengaktifkan *relay* yang telah disambungkan ke motor. Relay di supply dengan tegangan +5V. Sehingga motor dapat berjalan atau bekerja sesuai program. Gambar dari *wiring relay* dan motor beserta tabel keterangannya dapat dilihat pada Gambar 3.9 dan Tabel 3.4.



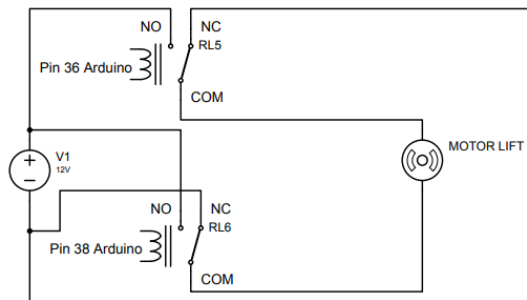
Gambar 3.9 Wiring Relay dan Motor

Tabel 3.4 Wiring Motor dan Relay

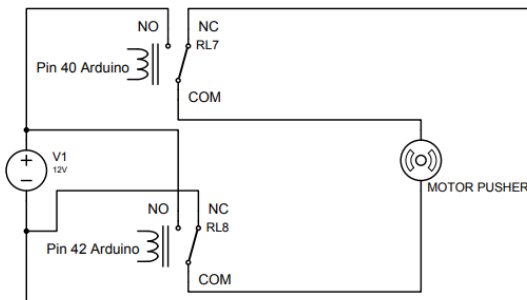
| No. | Konveyor | Motor | Wiring Motor | Relay | |
|-----|------------|-------|--------------|-------|-----|
| | | | | NO | COM |
| 1. | 2 | 2 | 14 | 2 | - |
| 2. | 3 | 3 | 15 | 3 | - |
| 3. | 4 | 4 | 16 | 4 | - |
| 4. | Lift naik | 5 | 17 | - | 5 |
| 5. | Lift turun | | 18 | - | 6 |

| No. | Konveyor | Motor | Wiring Motor | Relay | |
|-----|----------------------|-------|--------------|-------|-----|
| | | | | NO | COM |
| 6. | <i>Pusher</i> maju | 6 | 19 | - | 7 |
| 7. | <i>Pusher</i> mundur | 6 | 20 | - | 8 |

Pada Gambar 3.9 merupakan *wiring* dari motor dan *relay* dari motor yang digunakan untuk menggerakkan konveyor. *Wiring* antara konveyor 2, 3, 4 berbeda dengan *lift* dan *pusher*. Karena *lift* dan *pusher* berjalan ke kanan dan ke kiri, maka 1 motor dihubungkan dengan 2 *relay*. Berikut adalah rangkaian *lift* motor dengan *relay* pada Gambar 3.10 dan rangkaian *pusher* motor dengan *relay* pada Gambar 3.11.



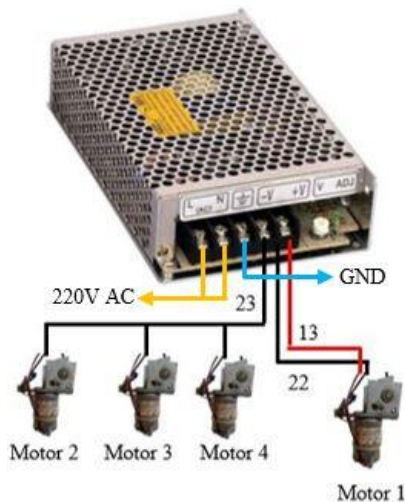
Gambar 3.10 Rangkaian *Relay* dan *Lift Motor*



Gambar 3.11 Rangkaian *Relay* dan *Pusher Motor*

3.3.5 Wiring Power Supply dan Motor

Barcode scanner terletak di ujung konveyor 1, sehingga konveyor 1 dibuat otomatis akan aktif ketika *push button ON* diaktifkan dan akan mati ketika tombol *OFF* ditekan. Karena konveyor 1 berjalan tanpa perlu adanya program maka motor 13 atau positif dari motor 1 disambungkan langsung ke V+, dan motor 22 sebagai negatif menerima *supply* dari V- *power supply*. Dan untuk konveyor 2, 3 dan 4 menggunakan 1 sumber negatif yaitu 23, dan disambungkan ke *supply* V-. *Power supply* di *supply* 220V AC. Gambar dan tabel dari *wiring power supply* dan motor dapat dilihat pada Gambar 3.12 dan Tabel 3.5.



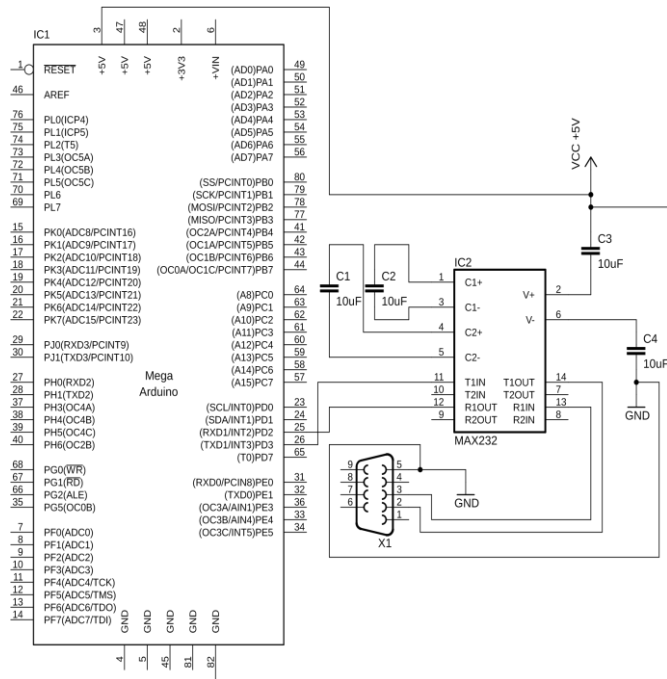
Gambar 3.12 *Wiring Power Supply* dan Motor

Tabel 3.5 *Wiring Power Supply* dan Motor

| No. | Konveyor | Motor | Power Supply | Motor |
|-----|----------|-------|--------------|-------|
| 1. | 1 | 1 | V+ | 13 |
| | | | V- | 22 |
| 2. | 2 | 2 | V- | 23 |
| 3. | 3 | 3 | | |
| 4. | 4 | 4 | | |

3.3.6 Wiring Barcode Scanner

Tipe *barcode* yang digunakan pada alat ini adalah ITF-6 karena mengingat ukuran konveyor yang tidak terlalu besar dan adanya sensor inframerah di konveyor, sehingga ukuran dari barang dibuat tidak terlalu besar agar dapat berjalan dengan baik ketika pengelompokan. Dengan ukuran barang yang kecil tersebut, maka ukuran *barcode* yang awalnya akan menggunakan tipe ITF-14 menjadi ITF-6 agar *scanner* dapat mudah mendeteksi *barcode* yang berjalan di konveyor. Sebelum membuat program untuk mendeteksi *barcode* tersebut, terlebih dahulu dibuat rancangan *hardware*-nya. Rangkaiannya dapat dilihat pada Gambar 3.13.



Gambar 3.13 Wiring Max232, RS232 DB 9 Female

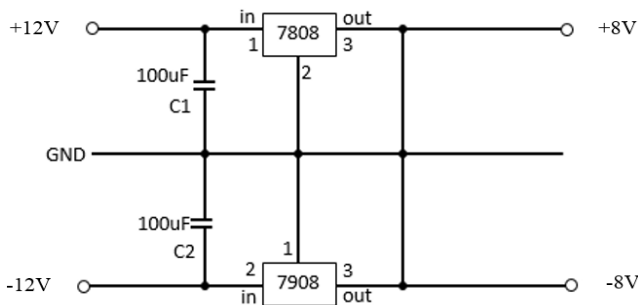
Untuk lebih jelasnya *wiring* antara Pin Max232, RS232, DB 9 Female dan Pin pada Arduino Mega 2560 dapat dilihat pada Tabel 3.6.

Tabel 3.6 Wiring Komunikasi Serial dan Arduino Mega 2560

| No. | IC MAX232 | Sambungkan ke |
|-----|-----------------|------------------------------|
| 1 | Pin 1 (C1+) | Capasitor 1 (+) |
| 2 | Pin 2 (V+) | Capasitor 4 (+) |
| 3 | Pin 3 (C1-) | Capasitor 1 (-) |
| 4 | Pin 4 (C2+) | Capasitor 2 (+) |
| 5 | Pin 5 (C2-) | Capasitor 2 (-) |
| 6 | Pin 6 (V-) | Capasitor 3 (-) |
| 7 | Pin 11 (T1 in) | Pin 18 (TX) Arduino |
| 8 | Pin 12 (R1 out) | Pin 19 (RX) Arduino |
| 9 | Pin 13 (R1 in) | Pin 3(RXD) DB9 Female |
| 10 | Pin 14 (T1 out) | Pin 2(TXD) DB9 Female |
| 11 | Pin 15 (GND) | Pin 5(GND) DB9Female |
| 12 | Pin 16 (VCC) | +5V Arduino |
| 13 | | Capasitor 4 (-) ke GND |
| 14 | | Capasitor 3 (-) ke GND |
| 15 | | Pin 5(GND) DB9Female Ke GND) |

3.3.7 Rangkaian Voltage Regulator

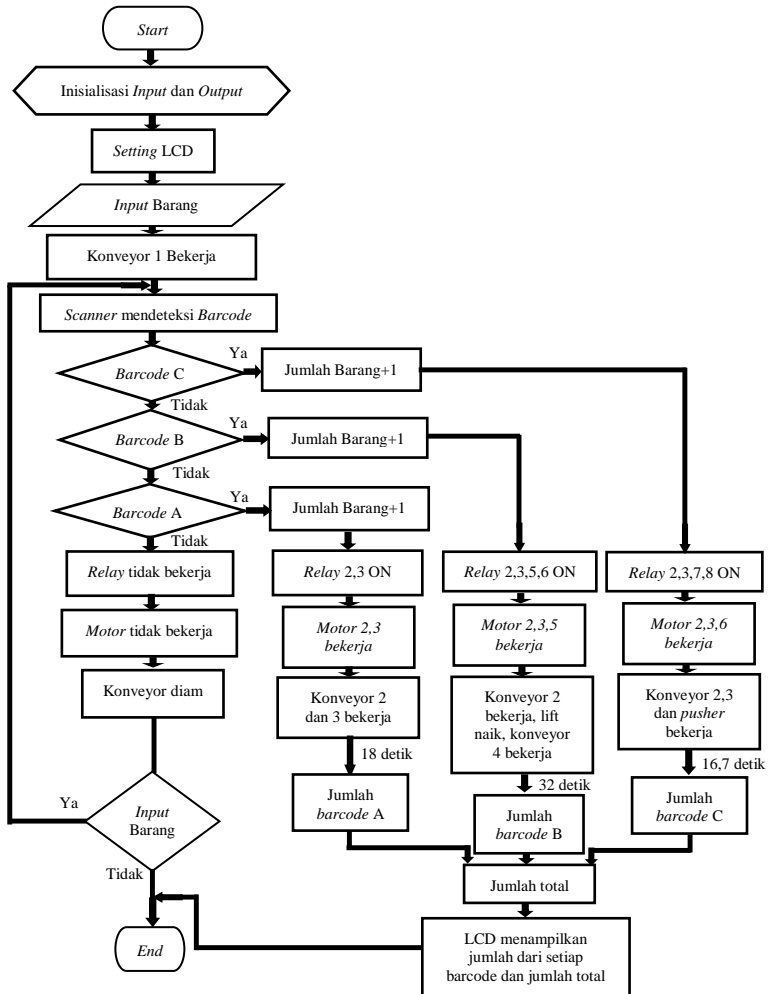
Rangkaian *voltage regulator* yang digunakan di alat ini yaitu menggunakan LM7808 sebagai *output* positif dan LM7908 sebagai *output* negatif. Rangkaian ini digunakan untuk *supply* motor konveyor 1, karena jika menggunakan *supply* 12V, jalannya konveyor terlalu cepat sehingga sensor susah untuk mendeteksi barang. Maka, *supply* 12V diturunkan menjadi 8V. Rangkaian *voltage regulator* yang digunakan dapat dilihat pada Gambar 3.14.



Gambar 3.14 Rangkaian Voltage Regulator

3.4 Perancangan Perangkat Lunak (*Software*)

Setelah merancang *hardware* beserta elektroniknya, maka dapat dibuat *software* agar alat dapat berjalan sesuai dengan yang diinginkan. *Flowchart* dapat dilihat pada Gambar 3.15.



Gambar 3.15 Flowchart

Pada Gambar 3.13, dapat dijelaskan bahwa ketika *start/mulai*, maka Arduino akan meng-inisialisasi *input* dan *output* yang digunakan, kemudian LCD akan menyala, dan konveyor 1 bekerja. Maka *input* barang dapat dilakukan. *Scanner barcode* akan mendeteksi termasuk *barcode* apakah itu. Ketika *barcode* yang terdeteksi adalah C maka Arduino akan menyimpan data tersebut dan menjumlahkan dengan data sebelumnya. *Relay* 2, 3, 7, 8 mengontak motor 2, 3, 6 sehingga konveyor 2 dan 3 akan berjalan kemudian *pusher* mendorong barang, setelah selesai, *pusher* kembali ke keadaan semula. Waktu untuk mengelompokkan *barcode* C adalah 20s, kemudian Arduino akan menyimpan data jumlah *barcode* C, lalu total keseluruhan *barcode* dijumlahkan, dan LCD akan menampilkan hasilnya. Jika *input* barang yang dideteksi tidak termasuk *barcode* A, B, C maka konveyor tidak akan bekerja/diam.

Penjabaran program utama pada Tugas Akhir ini adalah sebagai berikut.

- a. Program Inisialisasi LCD

```
#include <Wire.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd(19, 18, 17, 16, 15, 14);
```

Gambar 3.16 Program Inisialisasi Alamat pada LCD

Pada Gambar 3.16 terlihat inisialisasi alamat untuk LCD pada Arduino. Pin yang digunakan untuk LCD di Arduino yaitu Pin 14, 15, 16, 17, 18, 19.

- b. Program Inisialisasi *Scanner Barcode*

```
String inputString = "";
boolean stringComplete = false;
String barcode;
```

Gambar 3.17 Program Inisialisasi *Scanner Barcode*

Pada Gambar 3.17 terlihat inisialisasi awal untuk *scanner barcode*. *Barcode* menggunakan pin 18 (TX) dan pin 19 (RX) pada Arduino. *Barcode* akan mengirimkan data yang dikirim melalui komunikasi serial RS232 agar data diubah menjadi serial ketika dikirim ke Arduino Mega 2560.

c. Program Inisialisasi *Relay*

```
#define relayon 0
#define relayoff 1

int relay1 = 5;
int relay2 = 6;
int relay3 = 7;
int relay4 = 8;
int relay5 = 9;
int relay6 = 10;
int relay7 = 11;
int relay8 = 12;
```

Gambar 3.18 Program Inisialisasi *Relay*

Pada Gambar 3.18 merupakan program inisialisasi *relay* dengan Pin pada Arduino yang digunakan adalah Pin 5 hingga Pin 12.

-----Halaman ini sengaja dikosongkan-----

BAB IV

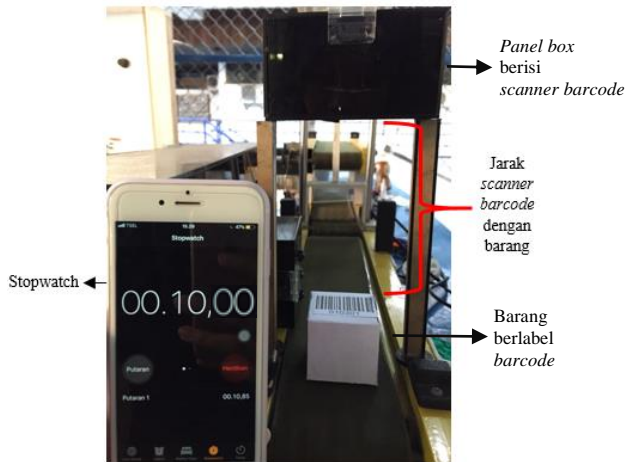
PENGUJIAN DAN ANALISA

Setelah perancangan dan pembuatan alat selesai, maka dapat dihubungkan antara *hardware* dan *software* alat sehingga menjadi sebuah sistem kerja. Kemudian dilakukan pengujian dan analisa data. Berikut pengukuran dan analisa data yang dilakukan.

4.1 Pengukuran Jarak MCR12 Me-scan Barcode

Sensor yang digunakan pada pengujian ini adalah *scanner barcode*. Tujuan dari pengujian *scanner barcode* ini adalah untuk mengetahui jarak pembacaan *scanner barcode* ketika barang diam. Pengujian ini dilakukan pada barang berlabel *barcode* ITF-6 seperti Gambar 4.1. Dan pengujian ini menggunakan *scanner barcode* MCR12 dan Arduino Mega saja. Data yang diambil pada pengujian sensor ini diantaranya adalah banyak *barcode* yang ter-scan dengan jarak dan waktu yang telah ditentukan.

Langkah-langkah pengujian ini yaitu dengan cara merangkai dan memprogram *scanner barcode* terlebih dahulu. Kaki-kaki sensor disambungkan ke pin Arduino. Rangkaian *scanner barcode* dapat dilihat pada Gambar 3.13 halaman 29. Kemudian, *upload* program yang dapat dilihat pada Lampiran B halaman B-1. Arduino disambungkan oleh sumber, kemudian *scanner barcode* akan aktif. Selanjutnya, *input* barang berlabel *barcode* di konveyor dengan jarak yang telah ditentukan, dan hitung waktu pengujian yang diukur menggunakan *stopwatch* seperti Gambar 4.1. Dalam rentang waktu yang telah ditentukan, hitung *barcode* yang ter-scan dengan indikator bunyi “tit” yang menandakan *barcode* ter-scan satu kali. Kemudian, ketika melewati waktu yang telah ditentukan maka *stopwatch* di *off* kan. Hasil deteksi pengukuran jarak MCR12 me-scan *barcode* dapat dilihat pada Tabel 4.1 dan tabel jumlah rata-rata *barcode* ter-scan dapat dilihat pada Tabel 4.2.



Gambar 4.1 Pengukuran Jarak MCR12 Me-scan Barcode

Tabel 4.1 Jumlah Barcode Ter-scan

| Jarak (cm) | Waktu (detik) | Jumlah Barcode Ter-scan |
|------------|---------------|-------------------------|
| 10 | 10 | 9 |
| 10 | 20 | 13 |
| 10 | 30 | 24 |
| 15 | 10 | 9 |
| 15 | 20 | 16 |
| 15 | 30 | 20 |
| 20 | 10 | 10 |
| 20 | 20 | 20 |
| 20 | 30 | 30 |
| 25 | 10 | 7 |
| 25 | 20 | 20 |
| 25 | 30 | 29 |

Data pada Tabel 4.1 berguna untuk mengetahui berapa banyak *barcode* yang dapat ter-scan dalam selang waktu 10 sampai 30 detik dalam jarak 10cm, 15cm, 20cm, dan 25cm. Berikut adalah

tabel rata-rata *scanner* dapat me-scan *barcode* dalam jarak tertentu dan dengan waktu 10-30 detik.

Tabel 4.2 Jumlah Rata-Rata *Barcode* Ter-scan

| No. | Jarak (cm) | Jumlah Rata-Rata |
|-----|------------|------------------|
| 1. | 10 | 15 |
| 2. | 15 | 15 |
| 3. | 20 | 20 |
| 4. | 25 | 18 |

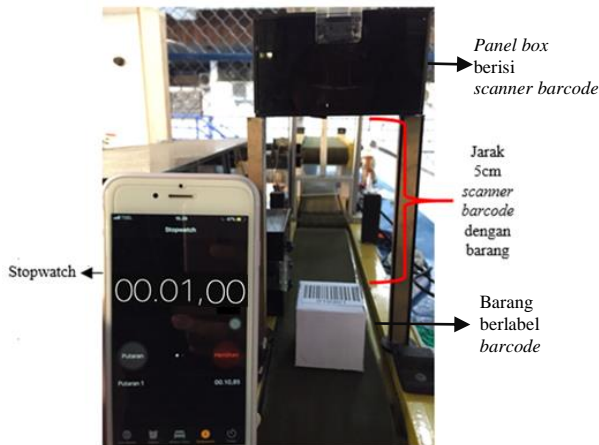
Berdasarkan Tabel 4.2, dapat disimpulkan bahwa *scanner barcode* dapat lebih mudah untuk mendeteksi *barcode* dengan jarak 20cm.

4.2 Pengukuran Kecepatan MCR12 Me-scan *Barcode*

Sensor yang digunakan pada pengujian ini adalah *scanner barcode*. Tujuan dari pengujian *scanner barcode* ini adalah untuk mengetahui kecepatan pembacaan *scanner barcode*. Pengujian ini dilakukan pada barang berlabel *barcode* ITF-6 dengan jarak 5cm, 10cm, 15cm, 20cm, 25cm dengan mengukur kecepatan waktu menggunakan *stopwatch* yang dapat dilihat pada Gambar 4.2 sampai Gambar 4.6. Pengujian ini menggunakan *scanner barcode* MCR12, konveyor, dan Arduino Mega. Data yang diambil pada pengujian sensor ini diantaranya adalah waktu ketika *barcode* ter-scan dengan jarak yang telah ditentukan.

Langkah-langkah pengujian ini yaitu dengan cara merangkai jalannya konveyor 1 menggunakan *voltage regulator* dan juga merangkai *scanner barcode*. Rangkaian *voltage regulator* disambungkan dengan konveyor 1 yang dapat dilihat pada Gambar 3.14 halaman 30. Kaki-kaki sensor disambungkan ke pin Arduino. Rangkaian *scanner barcode* dapat dilihat pada Gambar 3.13 halaman 29. Kemudian, *upload* program yang dapat dilihat pada Lampiran B halaman B-1. Aktifkan *push button*, maka konveyor 1 akan berjalan. Kemudian, sambungkan Arduino dengan sumber, maka *scanner barcode* akan aktif. Selanjutnya, *input* barang berlabel *barcode* di konveyor maka barang akan berjalan mendekati *panel box* yang berisi *scanner barcode*. Siapkan *stopwatch* dan aktifkan. Hentikan waktu ketika *scanner barcode* berbunyi “tit” yang menandakan *barcode* terdeteksi. Ulangi *input* barang di konveyor

dengan mengambil 5 data di setiap jarak. Hasil kecepatan waktu MCR12 *me-scan barcode* dapat dilihat pada Tabel 4.3 dan pengujian ketika *me-scan barcode* dengan jarak 5cm dapat dilihat pada Gambar 4.2.

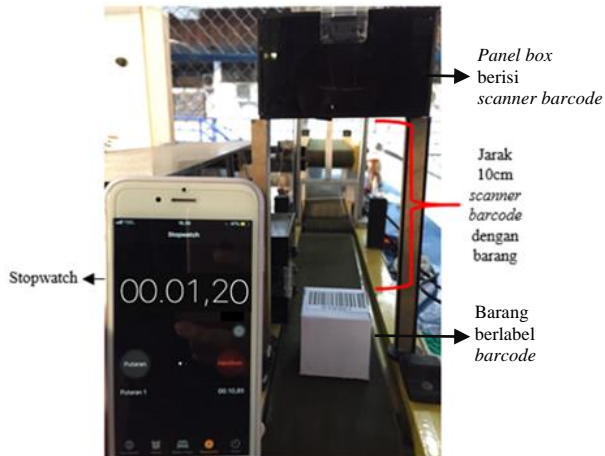


Gambar 4.2 Ketika *Me-scan Barcode* (5cm)

Tabel 4.3 Hasil Kecepatan Waktu MCR12 *Me-scan Barcode* (5cm)

| Jarak (cm) | Waktu (detik) |
|------------|---------------|
| 5 | 1 |
| 5 | 1 |
| 5 | 1,4 |
| 5 | 1,9 |
| 5 | 1,9 |

Berdasarkan Tabel 4.3, dapat disimpulkan bahwa dari 5 data pengujian tersebut, kecepatan rata-rata MCR12 *me-scan barcode* dengan jarak 5cm adalah 1,44 detik. Kemudian, dilakukan uji coba dengan jarak 10cm, berikut adalah gambar pengujian ketika *me-scan barcode* pada Gambar 4.3 dan hasil data pengukuran kecepatan MCR12 dengan jarak 10cm dapat dilihat pada Tabel 4.4.

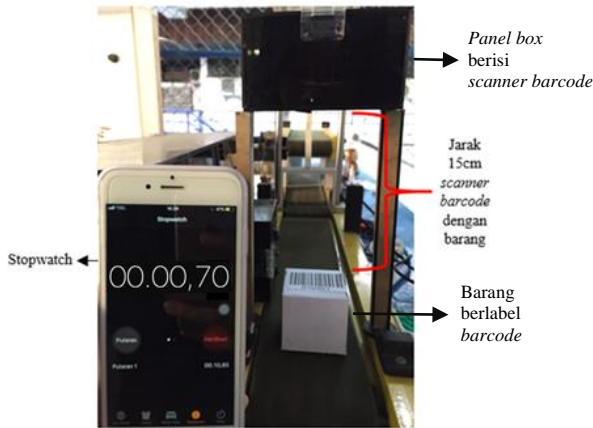


Gambar 4.3 Ketika Me-scan *Barcode* (10cm)

Tabel 4.4 Hasil Kecepatan Waktu MCR12 Me-scan *Barcode* (10cm)

| Jarak (cm) | Waktu (detik) |
|------------|---------------|
| 10 | 1,2 |
| 10 | 1,3 |
| 10 | 1,2 |
| 10 | 1 |
| 10 | 1,1 |

Berdasarkan Tabel 4.4, dapat disimpulkan dari 5 data pengujian tersebut, kecepatan rata-rata MCR12 me-scan *barcode* dengan jarak 10cm adalah 1,16 detik. Kemudian, dilakukan uji coba dengan jarak 15cm, berikut adalah gambar pengujian ketika me-scan *barcode* pada Gambar 4.4 dan hasil data pengukuran kecepatan MCR12 dengan jarak 15cm dapat dilihat pada Tabel 4.5.

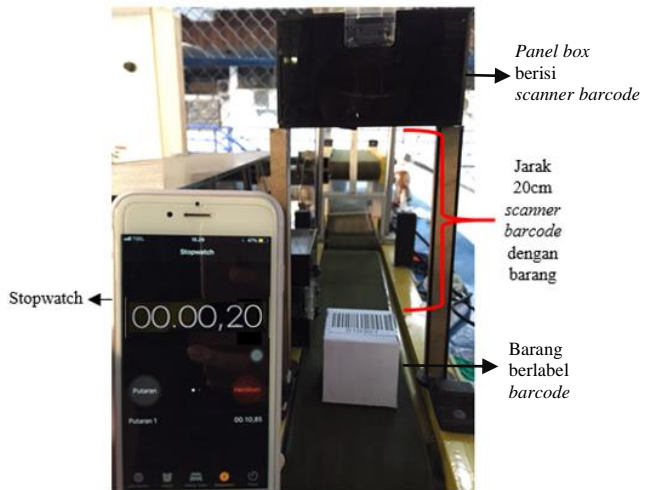


Gambar 4.4 Ketika Me-scan *Barcode* (15cm)

Tabel 4.5 Hasil Kecepatan Waktu MCR12 Me-scan *Barcode* (15cm)

| Jarak (cm) | Waktu (detik) |
|------------|---------------|
| 15 | 0,7 |
| 15 | 0,2 |
| 15 | 0,2 |
| 15 | 0,2 |
| 15 | 0,2 |

Berdasarkan Tabel 4.5, dapat disimpulkan bahwa dari 5 data pengujian tersebut, kecepatan rata-rata MCR12 me-scan *barcode* dengan jarak 15cm adalah 0,3 detik. Kemudian, dilakukan uji coba dengan jarak 20cm, berikut adalah gambar pengujian ketika me-scan *barcode* pada Gambar 4.5 dan hasil data pengukuran kecepatan MCR12 dengan jarak 20cm dapat dilihat pada Tabel 4.6.

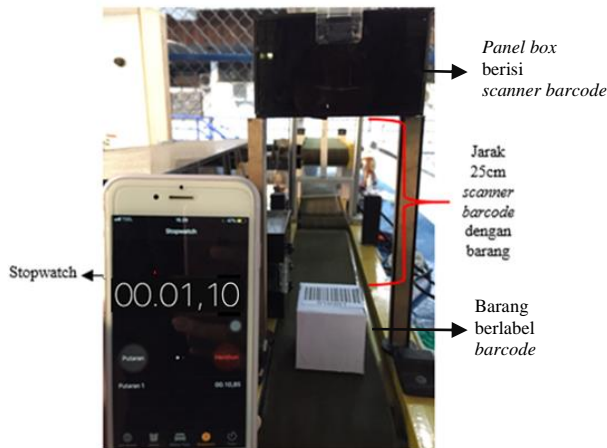


Gambar 4.5 Ketika Me-scan Barcode (20cm)

Tabel 4.6 Hasil Kecepatan Waktu MCR12 Me-scan Barcode (20cm)

| Jarak (cm) | Waktu (detik) |
|------------|---------------|
| 20 | 0,5 |
| 20 | 0,2 |
| 20 | 0,2 |
| 20 | 0,2 |
| 20 | 0,2 |

Berdasarkan Tabel 4.6 dapat disimpulkan bahwa dari 5 data pengujian tersebut, kecepatan rata-rata MCR12 me-scan barcode dengan jarak 20cm adalah 0,26 detik. Kemudian, dilakukan uji coba dengan jarak 25cm, berikut adalah gambar pengujian ketika me-scan barcode pada Gambar 4.6 dan hasil data pengukuran kecepatan MCR12 dengan jarak 25cm dapat dilihat pada Tabel 4.7.

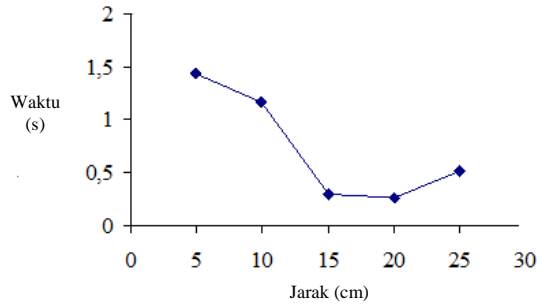


Gambar 4.6 Ketika Me-scan Barcode (25cm)

Tabel 4.7 Hasil Kecepatan Waktu MCR12 Me-scan Barcode (25cm)

| Jarak (cm) | Waktu (detik) |
|------------|---------------|
| 25 | 1,1 |
| 25 | 0,3 |
| 25 | 0,7 |
| 25 | 0,2 |
| 25 | 0,3 |

Berdasarkan Tabel 4.7, dari 5 data pengujian tersebut dapat disimpulkan bahwa kecepatan rata-rata MCR12 me-scan barcode dengan jarak 25cm adalah 0,52 detik. Dari beberapa data yang diambil, mengenai kecepatan ketika scanner dalam me-scan, maka dapat disimpulkan bahwa jarak yang tepat untuk scanner dalam me-scan atau mendeteksi barcode adalah 20cm yaitu dengan kecepatan 0,26 detik. Berikut adalah grafik hasil kecepatan waktu scanner barcode MCR12 dalam me-scan yang dapat dilihat pada Gambar 4.7.



Gambar 4.7 Hasil Kecepatan Waktu MCR12 Me-scan Barcode

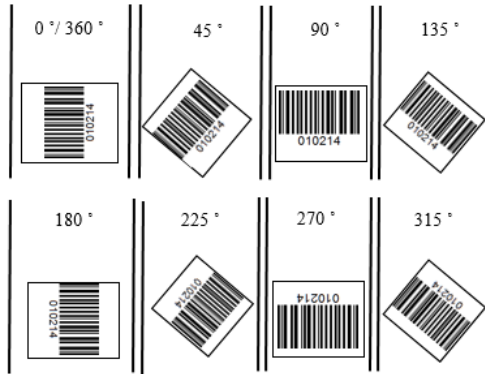
4.3 Pengujian Posisi Barcode

Untuk mendapatkan hasil yang maksimal ketika mendeteksi *barcode*, maka *barcode* ITF-6 pun diuji posisinya terhadap *scanner barcode* MCR12. Langkah-langkah pengujian posisi *barcode* yaitu dengan cara merangkai dan memprogram *scanner barcode* terlebih dahulu. Kaki-kaki sensor disambungkan ke pin Arduino. Rangkaian *scanner barcode* dapat dilihat pada Gambar 3.13 halaman 29. Kemudian, *upload* program yang dapat dilihat pada Lampiran B halaman B-1. Setelah *upload* program selesai, Arduino disambungkan dengan sumber, kemudian *scanner barcode* akan aktif. Selanjutnya, *input* barang berlabel *barcode* di konveyor dengan beberapa posisi sesuai derajat seperti Gambar 4.8. Ketika *scanner barcode* berbunyi “tit” yang menandakan *barcode* ter-scan, maka *scanner barcode* dapat mendeteksi dengan posisi tersebut. Ulangi percobaan dengan 9 posisi yang berbeda. Hasil pengujian posisi MCR12 me-scan *barcode* ITF-6 dapat dilihat pada Tabel 4.8.

Tabel 4.8 Hasil Pengujian Posisi Barcode ITF-6

| No. | Posisi (°) | Ket. |
|-----|------------|------------------|
| 1. | 0 ° | Tidak Terdeteksi |
| 2. | 45 ° | Tidak Terdeteksi |
| 3. | 90 ° | Terdeteksi |
| 4. | 135 ° | Tidak Terdeteksi |
| 5. | 180 ° | Tidak Terdeteksi |
| 6. | 225 ° | Tidak Terdeteksi |

| No. | Posisi (°) | Ket. |
|-----|------------|------------------|
| 7. | 270 ° | Terdeteksi |
| 8. | 315 ° | Tidak Terdeteksi |
| 9. | 360 ° | Tidak Terdeteksi |



Gambar 4.8 Pengujian Posisi Ketika Mendeteksi *Barcode*

Pada Tabel 4.8, dijelaskan bahwa keterangan dari beberapa pengujian adalah terdeteksi dan tidak terdeteksi. Arti dari terdeteksi ialah, *barcode* dengan posisi tersebut dapat terdeteksi oleh *scanner barcode*. Sedangkan, untuk keterangan tidak terdeteksi ialah, *barcode* dengan posisi tersebut tidak terdeteksi oleh *scanner barcode*. Dari 9 pengujian posisi tersebut, didapatkan bahwa *scanner barcode* dapat mendeteksi *barcode* dengan posisi 90 ° dan 270 °.

4.4 Tampilan LCD

Sensor yang digunakan pada pengujian ini adalah *scanner barcode*. Tujuan dari pengujian ini adalah untuk mengetahui tampilan LCD. Pengujian ini dilakukan pada 3 contoh barang berlabel *barcode* ITF-6 yang dapat dilihat pada Lampiran A contoh barang halaman A-2. Pengujian ini menggunakan *scanner barcode* MCR12, konveyor, Arduino Mega dan LCD.

Langkah-langkah pengujian ini yaitu dengan cara merangkai *relay* dengan Arduino seperti Gambar 3.7, merangkai *relay* dengan motor seperti Gambar 3.9, merangkai *power supply*

dengan *relay* seperti Gambar 3.8, merangkai *scanner barcode* seperti Gambar 3.13, dan LCD dengan Arduino seperti Gambar 3.6. Kemudian, *upload* program Arduino yang dapat dilihat pada Lampiran B halaman B-1. Aktifkan *push button*, maka konveyor 1 akan berjalan. Kemudian, sambungkan Arduino dengan sumber, maka *scanner barcode* akan aktif. Selanjutnya, *input* barang berlabel *barcode* di konveyor maka barang akan berjalan mendekati *panel box* yang berisi *scanner barcode*. *Scanner barcode* akan mendeteksi *barcode* kemudian mengelompokkannya sesuai barang yang telah ditentukan. Setelah proses pengelompokan selesai, maka LCD akan menampilkan hasil perhitungan dan pengelompokan yang dapat dilihat pada Gambar 4.9 sampai 4.12. LCD sebelum *barcode* di-*scan* dapat dilihat pada Gambar 4.9.



Gambar 4.9 LCD Sebelum *Scanner* Mendeteksi *Barcode*

Pada gambar 4.9 merupakan tampilan awal LCD dari jumlah *barcode* yang ter-*scan*, pada gambar terlihat jumlah dari keseluruhan barang adalah 0 karena belum mendeteksi *barcode*. Tampilan LCD ketika *barcode* mendeteksi barang dapat dilihat pada Gambar 4.10.



Gambar 4.10 LCD Setelah *Scanner* Mendeteksi *Barcode*

Pada Gambar 4.10 merupakan tampilan LCD ketika sedang proses mendeteksi barang dan mengelompokkan sesuai *barcode* yang akan dikelompokkan. Ketika barang terdeteksi maka konveyor akan melakukan proses sesuai yang diinginkan, setelah proses tersebut selesai maka tampilan pada LCD akan menampilkan hasil

dari pengelompokannya jumlah total dari keseluruhan barang di LCD akan bertambah. Berikut adalah tampilan di LCD ketika konveyor sudah mengelompokkan barang.



Gambar 4.11 LCD Setelah Konveyor Mengelompokkan Barang

Pada Gambar 4.11 merupakan tampilan LCD setelah mengelompokkan barang A, sehingga hasil dari pengelompokan dan jumlah dari keseluruhan barang bertambah. Tampilan di LCD ketika mendeteksi *barcode* lainnya dapat dilihat pada Gambar 4.12.



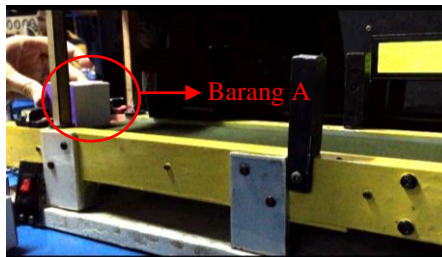
Gambar 4.12 LCD Setelah *Scanner* Mendeteksi Beberapa *Barcode*

Pada Gambar 4.12 merupakan tampilan LCD ketika mendeteksi beberapa *barcode* yang melewati konveyor. Setelah dikelompokkan, maka hasil pengelompokan dan jumlah dari keseluruhan barang bertambah.

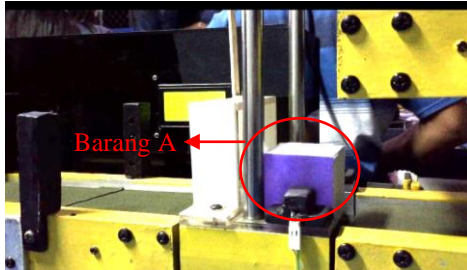
4.5 Pengujian Sistem Secara Keseluruhan

Setelah dilakukan pengujian pada *scanner barcode* dan LCD, kemudian dilakukan pengujian sistem perhitungan dan pengelompokan barang berdasarkan *barcode* ini pada konveyor. Pengujian sistem secara keseluruhan ini bertujuan untuk mengetahui kinerja keseluruhan dari alat ini. Pengujian dilakukan menggunakan 3 contoh barang berlabel *barcode* ITF-6 yang dapat dilihat pada Lampiran A contoh barang halaman A-2. Pengujian ini menggunakan *scanner barcode* MCR12, *relay*, konveyor, Arduino Mega dan LCD.

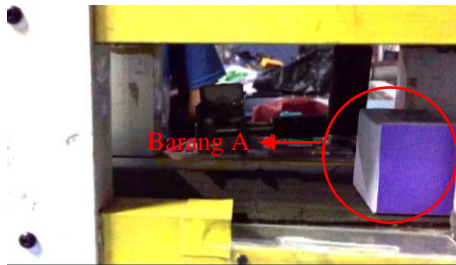
Langkah-langkah pengujian ini yaitu dengan cara merangkai *relay* dengan Arduino seperti Gambar 3.7, merangkai *relay* dengan motor seperti Gambar 3.9, merangkai *power supply* dengan *relay* seperti Gambar 3.8, merangkai *scanner barcode* seperti Gambar 3.13, dan LCD dengan Arduino seperti Gambar 3.6. Kemudian, *upload* program Arduino yang dapat dilihat pada Lampiran B halaman B-1. Aktifkan *push button*, maka konveyor 1 akan berjalan. Kemudian, sambungkan Arduino dengan sumber, maka *scanner barcode* akan aktif. Selanjutnya, *input* barang berlabel *barcode* di konveyor maka barang akan berjalan mendekati *panel box* yang berisi *scanner barcode*. *Scanner barcode* akan mendeteksi *barcode* kemudian mengelompokkannya sesuai barang yang telah ditentukan. Setelah proses pengelompokan selesai, maka LCD akan menampilkan hasil perhitungan dan pengelompokan. Kemudian, barang dapat di *input* kan kembali setelah proses pengelompokan selesai. Setelah *barcode* yang sama di *input* kan sebanyak 10 kali untuk pengujian, maka tekan tombol *reset* pada Arduino dan ambil data pengujian untuk *barcode* lainnya sebanyak 10 kali. *Step* dari pengelompokan barang dapat dilihat pada Gambar 4.13 sampai Gambar 4.27. Berikut adalah hasil dari pengujian sistem secara keseluruhan dengan pengujian setiap deteksi *barcode* yang dilakukan sebanyak 10 kali. Dan berikut ini adalah gambar *step* dari pengelompokan barang *barcode* A.



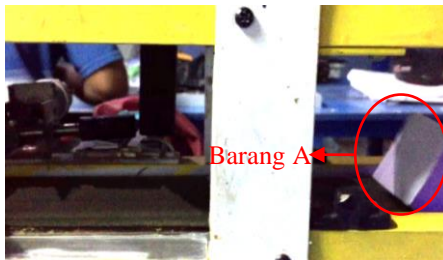
Gambar 4.13 *Scanner Barcode* Mendeteksi *Barcode* A



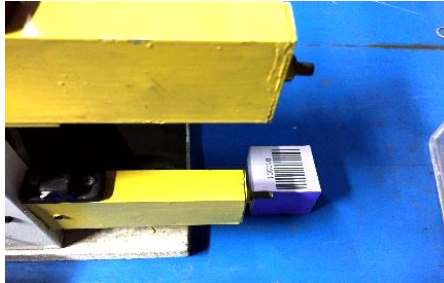
Gambar 4.14 Barang Melewati Konveyor 2



Gambar 4.15 Barang Melewati Konveyor 3



Gambar 4.16 Barang Sampai di Ujung Konveyor



Gambar 4.17 Kondisi Ketika *Barcode A* Dikelompokkan



Gambar 4.18 LCD Setelah Konveyor Mengelompokkan Barang A

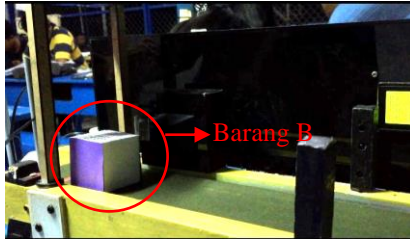
Pada Gambar 4.13, barang berjalan di konveyor 1 dan di konveyor tersebut barang di *scan*. *Scanner barcode* mendeteksi barang A, maka proses 1 akan berjalan. Barang berjalan melewati konveyor 2 seperti pada Gambar 4.14. Barang masih berjalan hingga melewati konveyor 3 hingga sampai di ujung konveyor kemudian dijatuhkan dari konveyor, yang ditunjukkan pada Gambar 4.15, Gambar 4.16, dan Gambar 4.17. Dan sesuai dengan Gambar 4.18, bahwa barang yang terdeteksi adalah *barcode A*. Berikut adalah pengujian deteksi *barcode A* dengan 10 kali percobaan yang dilakukan secara berurutan dapat dilihat pada Tabel 4.9.

Tabel 4.9 Hasil Pengujian *Barcode A*

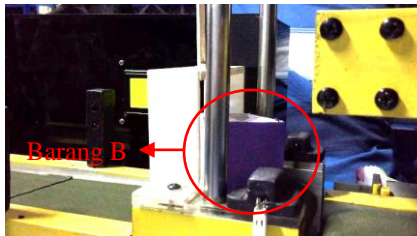
| No | <i>Barcode</i> | Terdeteksi | Proses | Waktu | Ket. |
|----|----------------|------------------|--------|----------|------------------|
| 1 | 010301 | Rendah | 6 | 21 detik | Tidak Terdeteksi |
| 2 | | <i>Barcode A</i> | 1 | 18 detik | Terdeteksi |
| 3 | | <i>Barcode A</i> | 1 | 15 detik | Terdeteksi |

| No | Barcode | Terdeteksi | Proses | Waktu | Ket. |
|----|---------|------------|--------|----------|------------------|
| 4 | 010301 | Barcode A | 1 | 20 detik | Terdeteksi |
| 5 | | Sedang | 5 | 34 detik | Tidak Terdeteksi |
| 6 | | Barcode A | 1 | 20 detik | Terdeteksi |
| 7 | | Rendah | 6 | 23 detik | Tidak Terdeteksi |
| 8 | | Rendah | 6 | 16 detik | Tidak Terdeteksi |
| 9 | | Rendah | 6 | 13 detik | Tidak Terdeteksi |
| 10 | | Barcode A | 1 | 17 detik | Terdeteksi |

Berdasarkan Tabel 4.9, didapatkan bahwa, ketika *barcode* A terdeteksi di konveyor 1 maka proses proses 1 akan berjalan. Proses 1 adalah kondisi ketika *relay* 2, 3 mengaktifkan motor pada konveyor 2 dan 3 yang kemudian barang akan berjalan lurus. Waktu rata-rata konveyor ketika mengelompokkan *barcode* A adalah 18 detik dari 5 data yang berhasil dideteksi. Berdasarkan Tabel 4.9, keterangan terdeteksi ialah ketika *scanner barcode* dapat mendeteksi dengan benar *barcode* yang ada di konveyor. Sedangkan, keterangan tidak terdeteksi ialah, ketika *barcode scanner* tidak mendeteksi adanya barang berlabel *barcode* atau salah mendeteksi *barcode* tersebut. Dari 10 percobaan tersebut, hanya 5 percobaan yang berhasil mendeteksi dengan benar karena *barcode* tidak terdeteksi oleh *scanner barcode* sehingga ketika melewati sensor inframerah barang terdeteksi untuk dikelompokkan sesuai ketinggian. Sensor inframerah dapat mendeteksi barang berlabel *barcode* walaupun barang berlabel *barcode* berwarna ungu, karena sensor inframerah dapat mendeteksi barang dengan indikator warna hitam atau putih. Karena warna barang untuk barang berlabel *barcode* berwarna ungu muda dan dikarenakan pencahayaan yang terlalu terang atau gelap, maka sensor inframerah dapat mendeteksi barang berlabel *barcode* yang berwarna ungu tersebut. Dari Tabel 4.9, dapat disimpulkan bahwa presentase *error* atau presentase ketika *scanner barcode* tidak mendeteksi *barcode* A adalah 50% dari 10 data yang diambil. Setelah pengujian *barcode* A selesai, maka tekan tombol *reset* pada Arduino, *input* kan barang berlabel *barcode* B sebanyak 10 kali secara bergantian. Berikut adalah pengujian *barcode* B beserta *step* pengelompokannya.



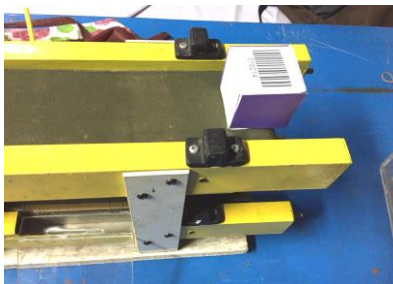
Gambar 4.19 *Scanner Barcode Mendeteksi Barcode B*



Gambar 4.20 *Barang Naik Lift*



Gambar 4.21 *Barang Melewati Konveyor 4*



Gambar 4.22 *Kondisi Ketika Barcode B Dikelompokkan*



Gambar 4.23 LCD Setelah Konveyor Mengelompokkan Barang B

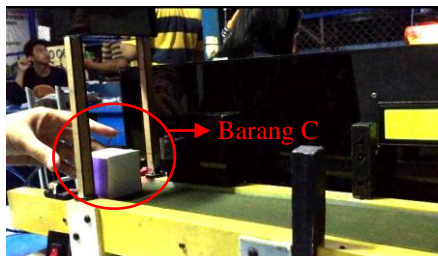
Pada Gambar 4.19, barang berjalan di konveyor 1 dan di konveyor tersebut barang di *scan*. *Scanner barcode* mendeteksi barang B, maka proses 2 akan berjalan. Barang berjalan ke konveyor 2 kemudian *lift* aktif seperti pada Gambar 4.20. Ketika *lift* berhenti, maka konveyor 2 akan aktif kemudian barang sampai di konveyor 4 seperti pada Gambar 4.21. Kondisi dari pengelompokan barang berlabel *barcode* B dapat dilihat pada Gambar 4.22. Dan sesuai dengan Gambar 4.23, bahwa barang yang terdeteksi adalah *barcode* B. Berikut adalah pengujian deteksi *barcode* B dengan 10 kali percobaan yang dilakukan secara berurutan dapat dilihat pada Tabel 4.10.

Tabel 4.10 Hasil Pengujian *Barcode* B

| No | Barcode | Terdeteksi | Proses | Waktu | Ket. |
|----|---------|------------|--------|----------|------------------|
| 1 | 010214 | Barcode B | 2 | 35 detik | Terdeteksi |
| 2 | | Barcode B | 2 | 34 detik | Terdeteksi |
| 3 | | Barcode B | 2 | 31 detik | Terdeteksi |
| 4 | | Barcode B | 2 | 30 detik | Terdeteksi |
| 5 | | Rendah | 2 | 17 detik | Tidak Terdeteksi |
| 6 | | Barcode B | 2 | 33 detik | Terdeteksi |
| 7 | | Barcode B | 2 | 31 detik | Terdeteksi |
| 8 | | Rendah | 2 | 16 detik | Tidak Terdeteksi |
| 9 | | Barcode B | 2 | 31 detik | Terdeteksi |
| 10 | | Barcode B | 2 | 31 detik | Terdeteksi |

Berdasarkan data pada Tabel 4.10, ketika *barcode* B terdeteksi di konveyor 1, maka proses 2 akan berjalan. Proses 2 adalah kondisi ketika *relay* 2 mengaktifkan motor konveyor 2

kemudian *relay* 5 aktif mengaktifkan *lift* untuk naik, lalu *relay* 2 dan 4 aktif untuk mengaktifkan konveyor 2 dan 4. Ketika barang sudah dikelompokkan, maka *relay* 6 akan aktif untuk mengaktifkan motor yang menurunkan *lift* kembali ke semula. Waktu rata-rata konveyor ketika mengelompokkan *barcode* B adalah 32 detik. Berdasarkan Tabel 4.10, keterangan terdeteksi ialah ketika *scanner barcode* dapat mendeteksi dengan benar *barcode* yang ada di konveyor. Sedangkan, keterangan tidak terdeteksi ialah, ketika *barcode scanner* tidak mendeteksi adanya barang berlabel *barcode* atau salah mendeteksi *barcode* tersebut. Dan dari 10 data hanya 8 data yang berhasil mendeteksi dengan benar karena *barcode* tidak terdeteksi oleh *scanner barcode* sehingga ketika melewati sensor inframerah barang terdeteksi untuk dikelompokkan sesuai ketinggian. Sensor inframerah dapat mendeteksi barang berlabel *barcode* walaupun barang berlabel *barcode* berwarna ungu, karena sensor inframerah dapat mendeteksi barang dengan indikator warna hitam atau putih. Karena warna barang untuk barang berlabel *barcode* berwarna ungu muda dan dikarenakan pencahayaan yang terlalu terang atau gelap, maka sensor inframerah dapat mendeteksi barang berlabel *barcode* yang berwarna ungu tersebut. Sehingga, presentase *error* dari pengujian *barcode* B adalah 20 % dari 10 percobaan yang dilakukan. Setelah pengujian *barcode* B selesai, maka tekan tombol *reset* pada Arduino, *input* kan barang berlabel *barcode* C sebanyak 10 kali secara bergantian. Berikut adalah pengujian *barcode* C beserta *step* pengelompokannya.



Gambar 4.24 *Scanner Barcode Mendeteksi Barcode C*



Gambar 4.25 *Pusher* mendorong barang



Gambar 4.26 Kondisi Ketika *Barcode C* Dikelompokkan



Gambar 4.27 LCD Setelah Konveyor Mengelompokkan Barang C

Pada Gambar 4.24, barang berjalan di konveyor 1 dan di konveyor tersebut barang di *scan*. *Scanner barcode* mendeteksi barang C, maka proses 3 akan berjalan. Barang berjalan melewati konveyor 2 dan 3. Kemudian, setelah sampai di *pusher*, *pusher* akan mendorong barang seperti Gambar 4.25. Kondisi akhir ketika barcode C dikelompokkan dapat dilihat pada Gambar 4.26. Dan sesuai dengan Gambar 4.27, bahwa barang yang terdeteksi adalah

barcode C. Berikut adalah pengujian deteksi *barcode* C dengan 10 kali percobaan yang dilakukan secara berurutan dapat dilihat pada Tabel 4.11.

Tabel 4.11 Hasil Pengujian *Barcode C*

| No | <i>Barcode</i> | Terdeteksi | Kondisi | Waktu | Ket. |
|----|----------------|------------------|---------|----------|------------------|
| 1 | 010315 | <i>Barcode C</i> | 3 | 12 detik | Terdeteksi |
| 2 | | <i>Barcode C</i> | 3 | 17 detik | Terdeteksi |
| 3 | | <i>Barcode C</i> | 3 | 18 detik | Terdeteksi |
| 4 | | <i>Barcode C</i> | 3 | 16 detik | Terdeteksi |
| 5 | | <i>Barcode C</i> | 3 | 19 detik | Terdeteksi |
| 6 | | Sedang | 5 | 36 detik | Tidak Terdeteksi |
| 7 | | Rendah | 6 | 16 detik | Tidak Terdeteksi |
| 8 | | <i>Barcode C</i> | 3 | 15 detik | Terdeteksi |
| 9 | | Rendah | 6 | 17 detik | Tidak Terdeteksi |
| 10 | | Rendah | 6 | 16 detik | Tidak Terdeteksi |

Ketika *barcode C* terdeteksi di konveyor 1, maka proses 3 akan berjalan. Proses 3 adalah kondisi dimana *relay* 2, 3 aktif untuk mengaktifkan konveyor 2 dan 3 kemudian *relay* 7 aktif ketika barang telah sampai di depan *pusher*, kemudian *pusher* akan mendorongnya. Setelah selesai, *relay* 8 akan aktif untuk mengembalikan *pusher* ke keadaan semula. Berdasarkan Tabel 4.11, keterangan terdeteksi ialah ketika *scanner barcode* dapat mendeteksi dengan benar *barcode* yang ada di konveyor. Sedangkan, keterangan tidak terdeteksi ialah, ketika *barcode scanner* tidak mendeteksi adanya barang berlabel *barcode* atau salah mendeteksi *barcode* tersebut. Waktu rata-rata konveyor ketika mengelompokkan *barcode C* adalah 16,7 detik dari 6 percobaan yang berhasil mendeteksi dengan benar karena *barcode* tidak terdeteksi oleh *scanner barcode* sehingga ketika melewati sensor inframerah barang terdeteksi untuk dikelompokkan sesuai ketinggian. Sensor inframerah dapat mendeteksi barang berlabel *barcode* walaupun barang berlabel *barcode* berwarna ungu, karena sensor inframerah dapat mendeteksi barang dengan indikator warna hitam atau putih. Karena warna barang untuk barang berlabel *barcode* berwarna ungu muda dan dikarenakan pencahayaan yang

terlalu terang atau gelap, maka sensor inframerah dapat mendeteksi barang berlabel *barcode* yang berwarna ungu tersebut. Sehingga, presentase *error* dari pengujian *barcode* C adalah 40 % dari 10 percobaan yang dilakukan.

Jadi, dalam 30 percobaan yang telah dilakukan ini terdapat beberapa *error* yaitu salah mendeteksi jenis barang yang seharusnya dikelompokkan sesuai *barcode* menjadi berdasarkan ketinggian dan masih adanya campur tangan manusia ketika barang tidak berada tepat pada konveyor yang seharusnya. Presentase *scanner barcode* dalam mendeteksi dengan benar secara keseluruhan adalah 64% sedangkan presentase *error*-nya adalah 36%.

BAB V

PENUTUP

Setelah dilakukan proses perencanaan, pembuatan, dan pengujian sensor serta dengan membandingkan dengan teori-teori penunjang, maka ditarik kesimpulan dan saran dari kegiatan yang telah dilakukan.

5.1 Kesimpulan

Dari tugas akhir yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut :

1. *Scanner barcode* dapat mendeteksi *barcode* dengan jarak 10-25cm, namun jarak yang terbaik adalah 20cm.
2. *Scanner barcode* dapat mendeteksi *barcode* dengan jarak 5-25cm, namun waktu tercepat *scanner* dalam mendeteksi *barcode* adalah 20cm.
3. *Scanner barcode* dapat mendeteksi *barcode* dengan posisi 90° atau 270° dalam rotasi horizontal.
4. Waktu rata-rata ketika proses pengelompokan *barcode* A, B, dan C adalah 18 detik, 32 detik, dan 16,7 detik.
5. Presentase keberhasilan *scanner barcode* dalam mendeteksi *barcode* adalah 64 %.
6. Ketika pengelompokan barang, terkadang barang berjalan terlalu cepat atau lambat di konveyor sehingga masih ada campur tangan manusia agar barang dapat dikelompokkan dengan baik.
7. Barang terlalu ringan sehingga mudah terguling ketika proses pengelompokan di konveyor.
8. Warna barang terlalu muda sehingga sensor inframerah mendeteksi barang berlabel *barcode* sebagai barang sesuai ketinggian.

5.2 Saran

1. Pada proyek selanjutnya akan lebih baik apabila hasil deteksi *barcode* ditampilkan di PC dengan ditambahkan keterangan tanggal dan waktu ketika barang terdeteksi di konveyor.
2. Ketika proses pengelompokan barang, sebaiknya jalannya konveyor lebih dipertimbangkan lagi dengan

berat dari contoh barang agar barang tidak mudah terguling ketika proses tersebut.

3. Dalam melakukan pemasangan pin *barcode* pada Arduino harus lebih teliti agar data tidak mengalami *blocking* dengan program lainnya, yang menyebabkan data tidak dapat terkirim.
4. Lebih teliti ketika memasukkan data *barcode* yang akan digunakan pada program sesuai dengan yang telah ditentukan.
5. Warna pada contoh barang yang digunakan dibuat lebih berwarna seperti merah, biru, dll agar sensor inframerah tidak mudah mendeteksi.

DAFTAR PUSTAKA

- [1] Pratama, Andrik Kurnia Adi dan Rahmat Bagus Prasajo. Mesin Pengelompok Barang Berdasarkan Ketinggian dan Konveyor Pengangkut Menggunakan PLC. **Tugas Akhir**. Program D3 Teknik Elektro FTI-ITS. Januari. 2015.
- [2] Hibatul, Wildan. Perencanaan Ulang Belt Conveyor A2 Pada PLTU PT. PJB PAITON Dengan Kapasitas 3500 Ton/Jam. **Tugas Akhir**. Program Studi D3 Teknik Mesin FTI_ITS, Juli 2013.
- [3] Lingga Dewi, Fadila. Alat Penghitung Jumlah Barang Menggunakan Barcode ITF-14. **Tugas Akhir**. Program D3 Teknik Elektro FTI-ITS. Juni. 2017.
- [4], **Motor DC dan Pengertiannya**, <https://teknikelektronika.com/pengertian-motor-dc-prinsip-kerja-motor-dc/>, 18 Juni 2018.
- [5] Kadir, Abdul. **Buku Pintar Pemograman Arduino**. MediaKom. Yogyakarta. 2015.
- [6] Hidayatullah, Syarif. Rancang Bangun Alat Penghitung Jumlah Produksi Semen Berbasis Data Logger Menggunakan Sensor Infrared. **Tugas Akhir**. Program D3 Teknik Elektro FTI-ITS. Juni. 2010.
- [7], **Rangkaian Interface Serial USART MAX232 Untuk Microcontroller**, <http://www.inzarsalfikar.com/2009/07/rangkaian-interface-serial-usart-max232.html>, 20 Juni 2018.
- [8], **Komunikasi Serial RS232**, www.insinyoer.com/prinsip-kerja-komunikasi-serial-rs232/
- [9], **Liquid Crystal Display**, <http://elektronika-dasar.web.id/lcd-liquid-crystal-display/>, 25 Juni 2018.

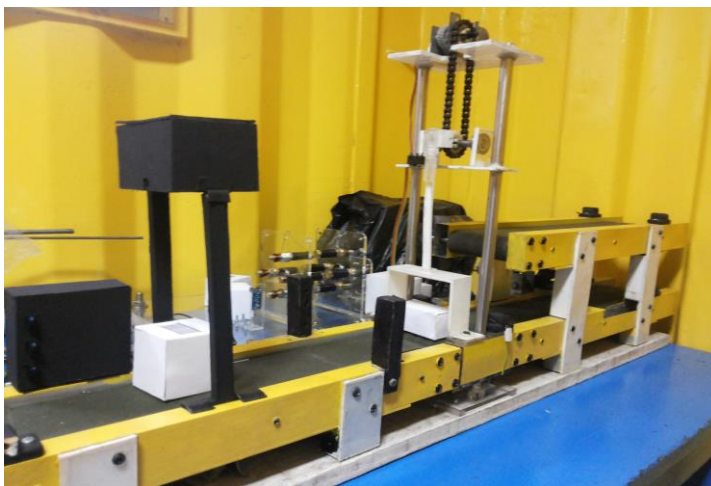
----Halaman ini Sengaja Dikosongkan----

LAMPIRAN A

A.1 Foto Alat Secara Keseluruhan



Gambar Alat Secara Keseluruhan



Gambar Peletakan *Scanner Barcode* di Konveyor

A.2 Contoh Barang Berlabel *Barcode*



Contoh Barang *Barcode* C



Contoh Barang *Barcode* B



Contoh Barang *Barcode* A

LAMPIRAN B

B.1 Pemrograman Arduino untuk Sensor

```
#include <Wire.h>
#include <LiquidCrystal.h>
#include <Ethernet.h>
#include <stdio.h>

LiquidCrystal lcd(23, 22, 17, 16, 15, 14);

byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
IPAddress ip(192, 168, 0, 175);
EthernetServer server(80);

#define relayon 0
#define relayoff 1

//String inputStringA = "00012345600012";    // a string to hold
incoming data
//String inputStringB = "12345678901217";
//String inputStringC = "30712345000010";

char stringA[8] = {'0', '1', '0', '3', '0', '1', '\r', 0};
char stringB[8] = {'0', '1', '0', '2', '1', '4', '\r', 0};
char stringC[8] = {'0', '1', '0', '3', '1', '5', '\r', 0};

char  inputString[14];

char lcd_buff[20]; //buffer karakter lcd

boolean stringCompleteA = false; // whether the string is complete
boolean stringCompleteB = false;
boolean stringCompleteC = false;
String barcode1, barcode2, barcode3;
int barcodeA = 0;
int barcodeB = 0;
int barcodeC = 0;
```

```

/*
int relay1 = 5;
int relay2 = 6;
int relay3 = 7;
int relay4 = 8;
int relay5 = 9;
int relay6 = 10;
int relay7 = 11;
int relay8 = 12;
*/

int relay1 = 28;
int relay2 = 30;
int relay3 = 32;
int relay4 = 34;
int relay5 = 36;
int relay6 = 38;
int relay7 = 40;
int relay8 = 42;

int h = 0, g = 0;
int n = 0;
char c;
char get_data[20];

char do_proses=0;

void setup()
{
  Serial.begin(115200);
  Serial1.begin(9600);
  Serial.println("INIT1");

  Ethernet.begin(mac, ip);
  server.begin();

  Serial.println("INIT2");
  //inputStringA.reserve(200);
  //inputStringB.reserve(200);

```

```

//inputStringC.reserve(200);
pinMode(relay1, OUTPUT);
pinMode(relay2, OUTPUT);
pinMode(relay3, OUTPUT);
pinMode(relay4, OUTPUT);
pinMode(relay5, OUTPUT);
pinMode(relay6, OUTPUT);
pinMode(relay7, OUTPUT);
pinMode(relay8, OUTPUT);

digitalWrite(relay1, relayoff);
digitalWrite(relay2, relayoff);
digitalWrite(relay3, relayoff);
digitalWrite(relay4, relayoff);
digitalWrite(relay5, relayoff);
digitalWrite(relay6, relayoff);
digitalWrite(relay7, relayoff);
digitalWrite(relay8, relayoff);

lcd.begin(16, 4);
lcd.setCursor(4,0);
lcd.print("MONITORING");
lcd.setCursor(4,1);
lcd.print("BARANG DI");
lcd.setCursor(5,2);
lcd.print("KONVEYOR");
delay(5000);

}

void serialEvent1() //get barcode
{
  char inA = 1;
  char inB = 1;
  char inC = 1;

  if (Serial1.available())
  { // If anything comes in Serial1 (pins 0 & 1)
    //Serial.write(Serial1.read());

```

```

c = (char)Serial1.read();

if (c == '\n')
{
    //Serial.println("printing");
    while (n >= 0)
    {
        if (inA == 1 && get_data[n] == stringA[n])
        {
            //Serial.write(stringA[n]);
            //Serial.println("cocok A");
        }
        else inA = 0;

        if (inB == 1 && get_data[n] == stringB[n])
        {

        } else inB = 0;

        if (inC == 1 && get_data[n] == stringC[n])
        {

        } else inC = 0;
        Serial.write(get_data[n]);
        n--;
    }
    if (inA == 1)
    {
        stringCompleteA = true;
        Serial.println("\nbarcode A");
    }
    if (inB == 1)
    {
        stringCompleteB = true;
        Serial.println("\nbarcode B");
    }
    if (inC == 1)
    {
        stringCompleteC = true;

```

```

        Serial.println("\nbarcode C");
    }
    n = 0;
    //Serial1.flush();
    while (Serial1.available() > 0)
    {
        char k = Serial1.read();
        //w++;
        delay(1);
    }
    Serial1.end();delay(100);
    Serial1.begin(9600);

}
else
{
    get_data[n] = c;
    n++;
}
}
}
void proses_decision()
{
    get_infrared();
    Serial.print("  flag_ir >>");
    Serial.println(flag_ir,DEC);

    if((stringCompleteA) && flag_set==0)
    {
        flag_set=1;
        do_proses=1;
    }
    else if((stringCompleteB) && flag_set==0)
    {
        //client.print("2");
        flag_set=1;
        do_proses=2;
    }
    else if((stringCompleteC) && flag_set==0)

```

```

{
  //client.print("3");
  flag_set=1;
  do_proses=3;
}
else if(flag_ir==1 && flag_set==0)
{
  flag_set=1;
  do_proses=4;
}
else if(flag_ir==2 && flag_set==0)
{
  flag_set=1;
  do_proses=5;
}
else if(flag_ir==3 && flag_set==0)
{
  flag_set=1;
  do_proses=6;
}
else
{
  //do_proses=0;
  //flag_set=0; // comment this line when connected to ethernet
client
}

flag_ir=0;

stringCompleteA=false;
stringCompleteB=false;
stringCompleteC=false;

Serial.print("proses decision> ");
Serial.println(do_proses,DEC);

delay(20);
}

```



```

void tampil_lcd()
{
    sprintf(lcd_buff,"Bar A:%2d Tgg:%2d",barcodeA,datatinggi); //
    untuk mengkopi string ke var lcd_buff
    lcd.setCursor(0, 0);
    lcd.print(lcd_buff);

    sprintf(lcd_buff,"Bar B:%2d Sdg:%2d",barcodeB,datasedang);
    lcd.setCursor(0, 1);
    lcd.print(lcd_buff);

    sprintf(lcd_buff,"Bar C:%2d Rnd:%2d",barcodeC,dataendah);
    lcd.setCursor(0, 2);
    lcd.print(lcd_buff);

    sprintf(lcd_buff,"Bar T:%2d
    Tot:%2d",barcodeC+barcodeB+barcodeA,dataendah+datasedang+d
    atatinggi);
    lcd.setCursor(0, 3);
    lcd.print(lcd_buff);
}

void loop()
{
    ///get_infrared();

    proses_decision();
    send2ethernet();
    tampil_lcd();

    if(flag_doProses>=100) //3dtk 60*50ms
    {
        proses(do_proses);
        do_proses=0;
        flag_set=0;
        flag_doProses=0;
        Serial.print("proses ");
    }
}

```

```

Serial.print(do_proses);
Serial.println("is done");
}

```

```

//Serial.println("main");
}

```

B.2 Pemrograman Arduino untuk Pengelompokan

```

void proses(int pr)
{
  if (pr==3||pr==6)
  {

    //Serial.println(inputStringA);
    //barcode1 = inputStringA.substring(0, 14);
    Serial.println("Deteksi Barcode A");
    Serial.print(barcode1);
    delay(1000);
    lcd.setCursor(14, 0);
    lcd.print(" ");
    //barcodeA++;
    Serial.print(barcode1);
    Serial.println(barcodeA);
    digitalWrite(relay2, relayon);
    delay(3000);
    digitalWrite(relay3, relayon);
    delay(500);
    digitalWrite(relay2, relayoff);
    delay(2000);
    digitalWrite(relay3, relayoff);
    delay(1000);
    digitalWrite(relay7, relayon);
    delay(2000);
    digitalWrite(relay7, relayoff);
    delay(1000);
    digitalWrite(relay8, relayon);

```

```

delay(2000);
digitalWrite(relay8, relayoff);
delay(1000);
//inputStringA = "00012345600012";
stringCompleteA = false;
Serial.println("A COMPLETED");

}

else if (pr==2||pr==5)
{

//Serial.println(inputStringB);
//barcode2 = inputStringB.substring(0, 14);
Serial.println("Deteksi Barcode B");
Serial.print(barcode2);
delay(1000);
lcd.setCursor(14, 0);
lcd.print(" ");
//barcodeB++;
Serial.print(barcode2);
Serial.println(barcodeB);
digitalWrite(relay2, relayon);
delay(1900);
digitalWrite(relay2, relayoff);
delay(3000);
digitalWrite(relay5, relayon);
delay(14800);
digitalWrite(relay5, relayoff);
delay(1000);
digitalWrite(relay2, relayon);
delay(1000);
digitalWrite(relay4, relayon);
delay(1200);
digitalWrite(relay2, relayoff);
delay(3000);
digitalWrite(relay4, relayoff);
delay(1000);
digitalWrite(relay6, relayon);

```

```

    delay(10900);
    digitalWrite(relay6, relayoff);
    delay(3000);
    //inputStringB = "12345678901217";
    stringCompleteB = false;
    Serial.println("B COMPLETED");
}

else if (pr==1||pr==4)
{
    //EthernetClient client = server.available();
    //client.print("PROSES C");

    //Serial.println(inputStringC);
    //barcode3 = inputStringC.substring(0, 14);
    Serial.println("Deteksi Barcode C");
    Serial.print(barcode3);
    delay(1000);
    lcd.setCursor(14, 0);
    lcd.print(" ");
    //barcodeC++;
    Serial.print(barcode3);
    Serial.println(barcodeC);
    digitalWrite(relay2, relayon);
    delay(2000);
    digitalWrite(relay3, relayon);
    delay(3500);
    digitalWrite(relay2, relayoff);
    delay(2000);
    digitalWrite(relay3, relayoff);
    delay(1000);
    //inputStringC = "30712345000010";
    stringCompleteC = false;
    Serial.println("C COMPLETED");
}

else
{
    digitalWrite(relay2, relayoff);

```

```

    digitalWrite(relay1, relayoff);
    digitalWrite(relay3, relayoff);
    digitalWrite(relay4, relayoff);
    digitalWrite(relay5, relayoff);
    digitalWrite(relay6, relayoff);
    digitalWrite(relay7, relayoff);
    digitalWrite(relay8, relayoff);
}

if(pr==1)barcodeA++;
else if(pr==2)barcodeB++;
else if(pr==3)barcodeC++;
else if(pr==4)datatinggi++;
else if(pr==5)datasedang++;
else if(pr==6)datarendah++;

flag_set=1;
}

```

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN C

C.1 Barcode Scanner MCR12

C.1.1 Datasheet MCR12

MCR12



Features

| | |
|---------------------|--|
| ● Scan Rate | 100 scans/sec $\pm 10\%$ |
| ● Depth Of Field | 230mm@20mil/0.5mm, PCS90% |
| ● Resolution | 5mil/0.127mm@PCS90% |
| ● Voltage | DC +5V $\pm 5\%$ |
| ● Power Consumption | 80mA |
| ● Connector | MOLEX 11P Pitch 1.25 |
| ● Interface | PS2/RS232/USB |
| ● Dimension | 44 mm W x 30 mm D x 19.2 mm H |
| ● Ambient light | 5000 Lux Max. |
| ● Advantage | Auto Sense , long reading range , small size |

C.1.2 *Specsheet* MCR12

| |
|----------------------|
| SPECIFICATION |
|----------------------|

Customer : _____

Customer's Model No. : _____

Model No. : **MCR12** _____

Date : _____

Sample Serial No. : _____

Spec. Version & Revision Date: V01 2010.07.19 _____

| |
|--|
| Received/Approved by |
| |



Web : <http://www.champtek.com>
E-mail : sales@champtek.com

ChampTek Incorporated
5/F, No. 2, Alley 2, Shih-Wei Lane, Chung Cheng Rd.,
Hsin Tien City, Taipei, Taiwan, R.O.C.
Tel : 886-2-22192385
Fax : 886-2-22192387

TABLE OF CONTENTS

| | |
|------------------------------------|---|
| A. General Description | 1 |
| B. Auto-sense Reading Mode..... | 2 |
| C. Physical Characteristics | 2 |
| D. Electrical Characteristics..... | 4 |
| E. Performance..... | 4 |
| F. Environmental | 4 |
| G. Readable Symbolologies | 5 |
| H. Pin Assignment | 6 |
| I. Scan Map | 8 |
| J. Reliability..... | 8 |

A. General Description

The MCR Series brings the benefits of bar code scanning to a variety of OEM devices.

The MCR Series Scan Module is a perfect choice for your OEM design. The MCR Series brings the benefits of bar code scanning to all types of OEM devices. Now kiosks, medical instruments, diagnostic equipment, lottery terminals, vending machines and countless other appliances can all be equipped with the leading-edge scanning technology and reliability.



The MCR Series has been designed to provide the highest scanning performance in the smallest package possible. For added versatility, allowing for fast, cost-effective interchangeability when upgrading or modifying your OEM device for specialized applications.

MCR12 is a compact long-range CCD bar code scanning module with high sensitive linear Image sensor and **built in Auto-sense function**. As remote-operation ability, 5,000 LUX ambient light resistance, 100 scan-rate per second and aiding in various major bar codes, hand-on software-programming function, this MCR12 is your best choice of CCD type bar code scanning module to be suitable for application where needs a high performance, small foot-print and the reliable operation.

MCR12 is a CCD bar code decoding capabilities. MCR12 decode board is powered by a fast processor and to decode a wide array of 1D bar codes. The decode board is compatible with Utility, a PC-based software for easy Configuration.

The MCR12 is designed with the industrial standard size, mounting options and output to facilitate integration into existing applications. The Scanner module's miniature size makes MCR12 ideal for integration into data terminals and other small devices. MCR12 is supplied as an assembled module with a mounting bracket or as separate components for custom mounting. The scanner module's unique open system architecture allows MCR12 to accept third party and custom plug-ins, giving the MCR12 virtually unlimited application flexibility.

Pb-Free (RoHS Compliant).

B. Auto-sense Reading Mode

Champtek's MCR12 with a futuristic design and high-performance scanning module represents the best auto-triggered scanner value on the OEM project.

The infrared light source can be sensitive automatically when barcode has been motion in front of MCR12 which means activating red light to scanning.

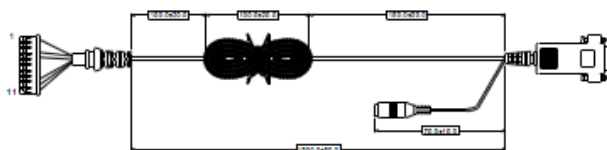
C. Physical Characteristics

| | |
|---------------------|-----------------------------------|
| Weight | |
| Body weight | Approx. 0.56 oz (17 g) |
| Cable weight(USB) | Approx. 1.87 oz (53 g) |
| Material | |
| Material | Polycarbonate |
| Cable Length | |
| Cable Length | 5FT. (150cm) |
| Connector | |
| Connector | WAFER 11P Pitch 1.25 |
| Dimension | |
| Dimension | 44.15 mm W x 31.95 mm D x 20 mm H |

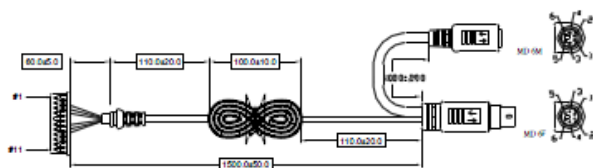
1) Cable drawing

Unit : mm

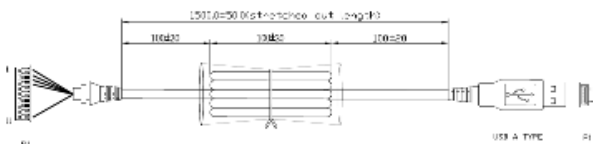
(1)RS232



(2) Keyboard

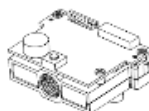
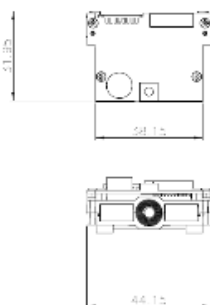


(3) USB



2) Mechanical drawing

Unit : mm



D. Electrical Characteristics

| Interface | RS232 | KB | USB |
|---------------------------|------------------|---------------|---------------|
| Supply Voltage | DC +5V $\pm 5\%$ | | |
| Output Voltage (Typ.) | $\pm 9V$ | +5V $\pm 5\%$ | +5V $\pm 5\%$ |
| Output low Voltage (Max.) | 0.7V | | |
| Current Draw | $\pm 10\%$ | | |
| Power On (Typ.) | 120mA | 120mA | 120mA |
| Stand by (Typ.) | 20mA | 20mA | 20mA |
| Operation (Typ.) | 100mA | 100mA | 115mA |
| Auto-sense standby (Typ.) | 80mA | 80mA | 80mA |

E. Performance

| | |
|----------------------|---|
| Light Source | Visible Red light 632nm LED |
| Sensor | Linear CCD Sensor |
| Processor Type | C8051 compatible |
| Operating Freq. | 24.5 MHz (Internal) |
| Scan Rate | 100 scans/sec $\pm 10\%$ |
| Reading Distance | 240mm@20mil/0.5mm, PCS90% |
| Print Contrast Ratio | PCS60% @ 6mil/0.15mm |
| Resolution | 5mil/0.127mm@PCS90% |
| Reading Angle | Test Conditions : Code 39, 10mil/0.25mm, PCS90% |
| Pitch Angle | 5°~70° ($\pm 5^\circ$) |
| Skew Tolerance | 5°~60° ($\pm 5^\circ$) |
| Ambient Light | 5000 Lux Max. |

F. Environmental

| | |
|-----------------------|-----------------------------------|
| Operating Temperature | 0 °C to 50 °C (32 °F to 122 °F) |
| Storage Temperature | -20 °C to 70 °C (-4 °F to 158 °F) |
| Relative Humidity | 20% to 95% (Non-condensing) |

G. Readable Symbolologies

| | Readable | Default Enable |
|-----------------------------|----------|----------------|
| All UPC/EAN/JAN | V | V |
| EAN128 Code | V | |
| Code 39 | V | V |
| Code 39 Full ASCII | V | |
| Code32 / Italian Pharmacy | V | |
| Code 128 | V | V |
| CODABAR/NW7 | V | V |
| Interleave 25 | V | V |
| Industrial 25 | V | |
| Matrix 25 | V | |
| MSI/PLESSEY | V | |
| Telepen | V | |
| Code 93 | V | |
| Code 11 | V | |
| China Postage | V | |
| Code 26 | V | |
| LCD25 | V | |
| GS1 DataBar Omnidirectional | V | |
| GS1 DataBar Limited | V | |
| GS1 DataBar Expanded | V | |

H. Pin Assignment

(a) RS232 Interface

DB 9 Female

| Pin No. | Function |
|------------|----------|
| 2 | TXD |
| 3 | RXD |
| 5 | GND |
| 7 | CTS |
| 8 | RTS |
| 9 | Vcc/+5V |
| Power Lead | Vcc/+5V |



(b) KBW Interface

DIN 5 MALE

| Pin No. | Function |
|---------|-----------|
| 1 | HOST CLK |
| 2 | HOST DATA |
| 4 | GND |
| 5 | Vcc(+5V) |



DIN 5 FEMALE

| Pin No. | Function |
|---------|----------|
| 1 | KB CLK |
| 2 | KB DATA |
| 4 | GND |
| 5 | Vcc(+5V) |



MiniDIN 6 MALE

| Pin No. | Function |
|---------|-----------|
| 1 | HOST DATA |
| 3 | GND |
| 4 | Vcc(+5V) |
| 5 | HOST CLK |



MiniDIN 6 FEMALE

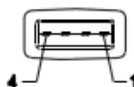
| Pin No. | Function |
|---------|----------|
| 1 | KB DATA |
| 3 | GND |
| 4 | Vcc(+5V) |
| 5 | KB CLK |



(c)USB Interface

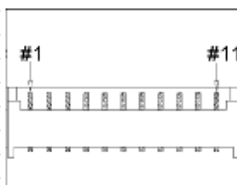
USB A TypeMale

| Pin No. | Function |
|---------|----------|
| 1 | Vcc |
| 2 | D- |
| 3 | D+ |
| 4 | GND |

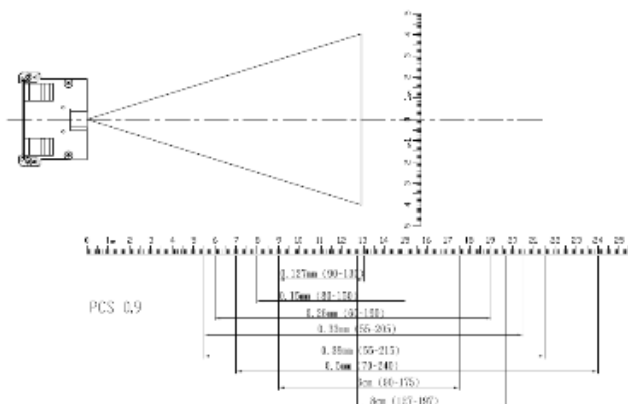


Decoder Data Output Connector

| Type | MOLEX (or Compatible) 11P Pitch 1.25 |
|---------|---|
| Pin No. | Function |
| 1 | GND |
| 2 | Vcc (+5V) |
| 3 | TXD |
| 4 | RXD |
| 5 | HOST DATA |
| 6 | HOST CLK |
| 7 | KB DATA |
| 8 | KB CLK |
| 9 | RTS |
| 10 | CTS |
| 11 | SHIELD |



I. Scan Map



1.3 Configuration MCR12

RS232 Mode Parameters

A> Set Up BAUD Rate

1200



2400



4800



<9600>



19200



38400



B> Set Up Data Bits

7 Data Bits



<8 Data Bits>



C> Set Up Stop Bits

<1 Stop>



2 Bits



D> Set Up Parity

<None>



Even



Odd



Mark



Space



E> Handshaking

RTS/CTS Enable



<RTS/CTS Disable>



ACK/NAK Enable



<ACK/NAK Disable>



XON/XOFF Enable



<XON/XOFF Disable>



Note: For more information, please visit www.champtek.com

Factory Default Settings

| Function | Default |
|-------------------------------|--|
| Interface | Keyboard Mode |
| Reading Mode | Guaranteed OFF |
| RS232 BAUD Rate | 9600 |
| Data Bits | Data Bits |
| Stop Bits | 1 Bit |
| Handshaking | RTS/CTS Disable |
| Keyboard Winge Mode | JBM PC(AT, PS/2) |
| Output Characters Transistor | CPH-E |
| Trans-out Selection Character | 0 ms |
| Symbologies | UPC-A UPC-E EAN-13/JAN-13 EAN-8/JAN-8 CODE 39 CODE 128 CODABAR/NW7 Interleaved 2 of 5 |
| Language | US English |
| Bar Code ID | OFF |
| Reading Level | Bar Equals Low |
| Accuracy | 1 Time |
| Beep Tone | High |
| Sensitivity of Continuous | Fast |
| Notebook Function | Disable |
| Reverse Output Character | Disable |

Warranty

Champtek Incorporated ("Champtek") manufactures its hardware products in accordance with industry standard practices. Champtek warrants that for a period of twenty-four (24) months from date of shipment, products will be free from defects in materials and workmanship.

This warranty is provided to the original owner only and is not transferable to any third party. Champtek will not repair any product which has been repaired or altered unless done or approved by Champtek, which has not been maintained in accordance with any operating or handling instructions supplied by Champtek, which has been subjected to unusual physical or electrical stress, misuse, abuse, power shortage, negligence or accident or which has been used other than in accordance with the product operating and handling instructions.

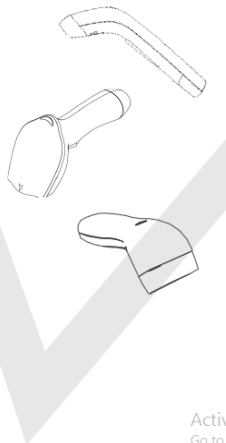
Champtek Incorporated
5/F, No. 2, Alley 2, Shih-Hsiung Lane, Chung-Cheng Rd.,
Hsin-Tien City, Taipei 231, Taiwan.
TEL: +886-2-2219-2365
FAX: +886-2-2210-2287
<http://www.champtek.com>



044-028501_V03

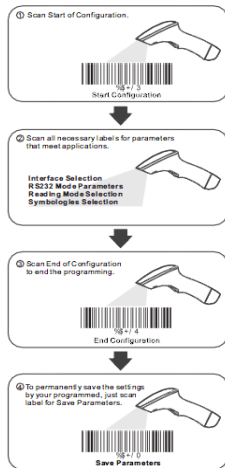
CHAMPTTEK®

Programming Quick Guide



Active
Go to P

Configuration Flow Chart



Set All Defaults



Version Information



Interface Selection



USB Mode



Reading Mode Selection



<Good Read OFF>



Continuous/Trigger OFF



Continuous/Auto Power On



Flash/Auto Power On



Select Terminator



<CR>LF>



CR



Space



STX/ETX



None



LF



HT(TAB)



Symbologies Selection

UPC-A <ON>



UPC-A OFF



UPC-E <ON>



UPC-E OFF



EAN-13/JAN-13/ISSN-13 <ON>



EAN-13/JAN-13/ISSN-13 OFF



EAN-8/JAN-8 <ON>



EAN-8/JAN-8 OFF



CODE 39 <ON>



CODE 39 OFF



CODE 128 <ON>



CODE 128 OFF



CODABAR/NW7 <ON>



CODABAR/NW7 OFF



Interleaved 2 of 5 <ON>



Interleaved 2 of 5 OFF



CODE 93 ON



CODE 93 OFF



China Postage ON



China Postage OFF



Telepen ON



Telepen OFF



GS1 DataBar Omnidirectional ON



GS1 DataBar Omnidirectional OFF



GS1 DataBar Limited ON



GS1 DataBar Limited OFF



GS1 DataBar Stacked ON



GS1 DataBar Stacked OFF



Select All Bar Codes



Select All Bar Codes



C.2 Datasheet IC MAX232



MAX232, MAX232I

31130170 – FEBRUARY 1989 – REVISED NOVEMBER 2014

MAX232x Dual EIA-232 Drivers/Receivers

1 Features

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0- μ F Charge-Pump Capacitors
- Operates up to 120 kbit/s
- Two Drivers and Two Receivers
- ± 30 -V Input Levels
- Low Supply Current: 8 mA Typical
- ESD Protection Exceeds JESD 22
 - 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1- μ F Charge-Pump Capacitors is Available With the MAX202 Device

3 Description

The MAX232 device is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept ± 30 -V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels.

Device Information⁽¹⁾

| ORDER NUMBER | PACKAGE (PIN) | BODY SIZE |
|--------------|---------------|---------------------------|
| MAX232x | SOIC (16) | 6.60 mm \times 3.81 mm |
| | SOIC (16) | 16.30 mm \times 7.50 mm |
| | POP (16) | 16.30 mm \times 6.35 mm |
| | SOP (16) | 16.3 mm \times 6.30 mm |

(1) For all available packages, see the orderable addendum at the end of the datasheet.

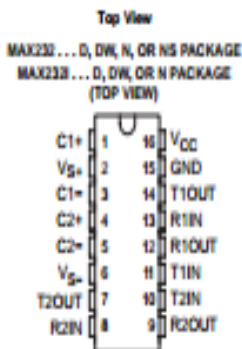
2 Applications

- TIA/EIA-232-F
- Battery-Powered Systems
- Terminals
- Modems
- Computers

4 Simplified Schematic



6 Pin Configuration and Functions



Pin Functions

| PIN | | TYPE | DESCRIPTION |
|--------------|--------|------|--|
| NAME | NO. | | |
| C1+ | 1 | — | Positive lead of C1 capacitor |
| VS+ | 2 | O | Positive charge pump output for storage capacitor only |
| C1- | 3 | — | Negative lead of C1 capacitor |
| C2+ | 4 | — | Positive lead of C2 capacitor |
| C2- | 5 | — | Negative lead of C2 capacitor |
| VS- | 6 | O | Negative charge pump output for storage capacitor only |
| T2OUT, T1OUT | 7, 14 | O | RS232 line data output (to remote RS232 system) |
| R2IN, R1IN | 8, 13 | I | RS232 line data input (from remote RS232 system) |
| R2OUT, R1OUT | 9, 12 | O | Logic data output (to UART) |
| T2IN, T1IN | 10, 11 | I | Logic data input (from UART) |
| GND | 15 | — | Ground |
| VCC | 16 | — | Supply Voltage, Connect to external 5V power supply |

7 Specifications

7.1 Absolute Maximum Ratings⁽¹⁾

over operating free-air temperature range (unless otherwise noted)

| | | MIN | MAX | UNIT |
|----------|---|----------------------------------|-------------------------------|------|
| V_{CC} | Input supply voltage range ⁽²⁾ | -0.3 | 6 | V |
| V_{OH} | Positive output supply voltage range | $V_{CC} - 0.3$ | 15 | V |
| V_{OL} | Negative output supply voltage range | -0.3 | -15 | V |
| V_I | Input voltage range | $T1IN, T2IN$ $R1IN, R2IN$ | -0.3 $V_{CC} + 0.3$ | V |
| V_O | Output voltage range | $T1OUT, T2OUT$ $R1OUT, R2OUT$ | $V_{OH} - 0.3$ $V_{OL} + 0.3$ | V |
| | Short-circuit duration | $T1OUT, T2OUT$ | Unlimited | |
| T_J | Operating virtual junction temperature | | 150 | °C |

- (1) Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under Recommended Operating Conditions is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
- (2) All voltages are with respect to network GND.

7.2 Handling Ratings

| | | MIN | MAX | UNIT |
|-----------|--|-----|------|------|
| T_{stg} | Storage temperature range | -65 | 150 | °C |
| V_{ESD} | Electrostatic discharge | | | V |
| | Human body model (HBM), per ANSI/ESDA/JEDEC JS-001, all pins ⁽¹⁾ | 0 | 2000 | |
| | Charged device model (CDM), per JEDEC specification JESD22-C101, all pins ⁽²⁾ | 0 | 1000 | |

- (1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.
- (2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

7.3 Recommended Operating Conditions

| | | MIN | NOM | MAX | UNIT |
|--------------|---|-------------------|----------|----------|------|
| V_{CC} | Supply voltage | 4.5 | 5 | 5.5 | V |
| V_{IH} | High-level input voltage ($T1IN, T2IN$) | 2 | | | V |
| V_{IL} | Low-level input voltage ($T1IN, T2IN$) | | | 0.8 | V |
| $R1IN, R2IN$ | Receiver input voltage | | | ± 30 | V |
| T_A | Operating free-air temperature | MAX232 MAX232I | 0 -40 | 70 85 | °C |

7.4 Thermal Information

| THERMAL METRIC ⁽¹⁾ | MAX232D | MAX232DWR | MAX232N | MAX232N/S | UNIT |
|-------------------------------|---------|-----------|---------|-----------|------|
| | SOIC | SOIC wide | PDIP | SOIC | |
| $R_{\theta JA}$ | 16 PINS | 16 PINS | 16 PINS | 16 PINS | °C/W |
| | 73 | 57 | 67 | 64 | |

- (1) For more information about traditional and new thermal metrics, see the IC Package Thermal Metric application report (SPRA953).

7.5 Electrical Characteristics — Device

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Figure 6)

| PARAMETER | TEST CONDITIONS ⁽¹⁾ | MIN | TYP ⁽²⁾ | MAX | UNIT |
|-----------|--------------------------------|---|--------------------|-----|------|
| I_{CC} | Supply current | $V_{CC} = 5.5V$, all output open, $T_A = 25^\circ C$ | | 8 | mA |

- (1) Test conditions are $C1-C4 = 1 \mu F$ at $V_{CC} = 5 V \pm 0.5 V$
- (2) All typical values are at $V_{CC} = 5 V$, and $T_A = 25^\circ C$.

7.6 Electrical Characteristics — Driver

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

| PARAMETER | TEST CONDITIONS ⁽¹⁾ | MIN | TYP ⁽²⁾ | MAX | UNIT |
|--|---|-----|--------------------|-----|---------------|
| V_{OH} High-level output voltage | T ₁ OUT, T ₂ OUT $R_L = 3\text{ k}\Omega$ to GND | 5 | 7 | | V |
| V_{OL} Low-level output voltage ⁽³⁾ | T ₁ OUT, T ₂ OUT $R_L = 3\text{ k}\Omega$ to GND | –7 | –5 | | V |
| t_D Output resistance | T ₁ OUT, T ₂ OUT $V_{DD} = V_{SS} = 5\text{ V}$, $V_O = 4.2\text{ V}$ | 300 | | | Ω |
| $I_{SD}^{(4)}$ Short-circuit output current | T ₁ OUT, T ₂ OUT $V_{DD} = 5.5\text{ V}$, $V_O = 0\text{ V}$ | | | ±10 | mA |
| I_{IS} Short-circuit input current | T ₁ IN, T ₂ IN $V_I = 0$ | | | 200 | μA |

(1) Test conditions are C₁–C₄ = 1 μF at $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$.

(2) All typical values are at $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$.

(3) The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

(4) Not more than one output should be shorted at a time.

7.7 Electrical Characteristics — Receiver

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

| PARAMETER | TEST CONDITIONS ⁽¹⁾ | MIN | TYP ⁽²⁾ | MAX | UNIT |
|---|--|-----|--------------------|-----|------------|
| V_{OH} High-level output voltage | R ₁ OUT, R ₂ OUT $I_{OH} = -1\text{ mA}$ | 3.5 | | | V |
| V_{OL} Low-level output voltage ⁽³⁾ | R ₁ OUT, R ₂ OUT $I_{OL} = 3.2\text{ mA}$ | | | 0.4 | V |
| V_{IT+} Receiver positive-going input threshold voltage | R ₁ IN, R ₂ IN $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$ | 1.7 | 2.4 | | V |
| V_{IT-} Receiver negative-going input threshold voltage | R ₁ IN, R ₂ IN $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$ | 0.8 | 1.2 | | V |
| V_{IH} Input hysteresis voltage | R ₁ IN, R ₂ IN $V_{CC} = 5\text{ V}$ | 0.2 | 0.5 | 1 | V |
| r_i Receiver input resistance | R ₁ IN, R ₂ IN $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$ | 3 | 5 | 7 | k Ω |

(1) Test conditions are C₁–C₄ = 1 μF at $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$.

(2) All typical values are at $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$.

(3) The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

7.8 Switching Characteristics

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

| PARAMETER | TEST CONDITIONS ⁽¹⁾ | MIN | TYP ⁽²⁾ | MAX | UNIT |
|---|---|-----|--------------------|-----|------------------|
| SR Driver slew rate | $R_L = 3\text{ k}\Omega$ to 7 k Ω , see Figure 4 | | | 30 | V/ μs |
| SR(tr) Driver transition region slew rate | see Figure 5 | | 3 | | V/ μs |
| Data rate | One TOUT switching | | 120 | | kbps |
| t_{PLH} Receiver propagation delay time, low- to high-level output | TTL load, see Figure 3 | | 500 | | ns |
| t_{PLHL} Receiver propagation delay time, high- to low-level output | TTL load, see Figure 3 | | 500 | | ns |

(1) Test conditions are C₁–C₄ = 1 μF at $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$.

C.3 Datasheet RS232 DB9 Female



Document Reference No.: FT_006204
DB9-USB-RS232 Datasheet
Version 1.2
Clearance No.: FTD1# 130

2.1.3 Replacing an RS232 DB9 FEMALE (DCE defined) Connector

The DB9-USB-RS232-F can be used to replace a female DB9 connector used for transmitting RS232 protocol. With the DB9-USB-RS232-F in place instead of the standard USB connector a USB bridge is created, this allow the application to communicate with other devices via USB. Installing the DB9-USB-RS232-F is simple. Simply replace the female DB9 connector with the DB9-USB-RS232-F connector (same PCB footprint), install drivers and the device is ready to use.

Table 2.3 gives the pin out description of each pad of an RS232 footprint. Figure 2.3 gives a description of the connections between the DB9-USB-RS232-F and the footprint of a female DB9 module. Figure 2.4 illustrates the connection made to an example PCB.

| Pin Number | Pin Type of DB9-USB-RS232 module | Pin Type at application PCB | Description |
|------------|----------------------------------|-----------------------------|---|
| 1 | Input | Output | DCD = Data Carrier Detect (this is an input to the DB9-USB-RS232-F from the application DCD output) |
| 2 | Input | Output | RXD = Receive Data (this is an input to the DB9-USB-RS232-F from the application Tx output, normally labelled RXD in DCE convention) |
| 3 | Output | Input | TXD = Transmit Data (this is an output to the DB9-USB-RS232-F from the application Rx input, normally labelled TXD in DCE convention) |
| 4 | Output | Input | DTR = Data Terminal Ready (this is an output to the DB9-USB-RS232-F from the application DTR input, normally labelled DTR in DCE convention) |
| 5 | Ground | Ground | GND = RS232 signal ground |
| 6 | Input | Output | DSR = Data Set Ready (this is an input to the DB9-USB-RS232-F from the application DTR output, normally labelled DSR in DCE convention) |
| 7 | Output | Input | RTS = Request To Send (this is an output to the DB9-USB-RS232-F from the application CTS input, normally labelled RTS in DCE convention) |
| 8 | Input | Output | CTS = Clear To Send (this is an input to the DB9-USB-RS232-F from the application RTS output, normally labelled CTS in DCE convention) |
| 9 | Input | Output | RI = Ring Indicator (this is an input to the DB9-USB-RS232-F from the application RI output) |
| Shield | Ground | Case Ground | Drain = typically connected to the host PC case |

Table 2.3 – A Female RS232 DB9 footprint Pin-Out

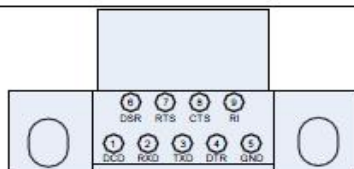


Figure 2.3 – DB9-USB-RS232-F Pin-Out from a Top View through the module

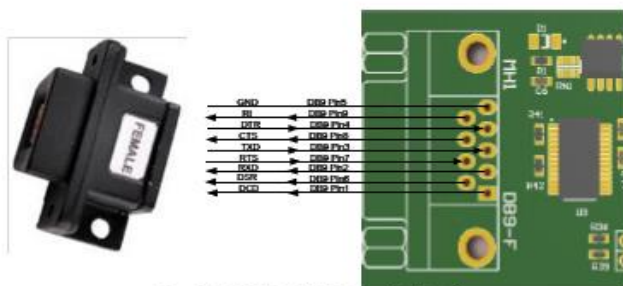


Figure 2.4 – DB9-USB-RS232-F Connection illustration



LM7800 Series 3-Terminal Fixed Voltage Regulators

THREE-TERMINAL POSITIVE VOLTAGE REGULATORS

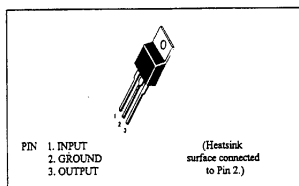
These voltage regulators are monolithic integrated circuits designed as fixed-voltage regulators for a wide variety of applications including local, on-card regulation. These regulators employ internal current limiting, thermal shutdown, and safe-area

FEATURES

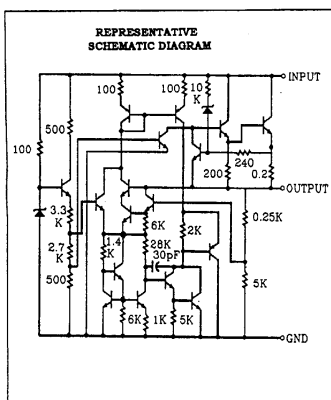
- Output Current in Excess of 1.5 Ampere
- No External Components Required
- Internal Thermal Overload Protection
- Internal Short-Circuit Current Limiting
- Output Transistor Safe-Area Compensation
- Output Voltage Offered in 2% Tolerance

compensation. With adequate heatsinking they can deliver output currents in excess of 1.5 ampere.

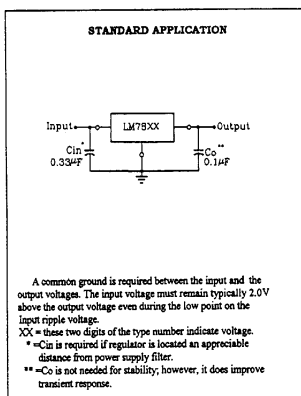
Although designed primarily as a fixed voltage regulator, these devices can be used with external components to obtain adjustable voltages and currents.

PIN ARRANGEMENT

CIRCUIT SCHEMATIC



TYPICAL CONNECTING CIRCUIT



LM7808 ELECTRICAL CHARACTERISTICS

($V_{in}=14V$, $I_{out}=500mA$, $0^{\circ}C \leq T_J \leq 125^{\circ}C$, $C_{in}=0.33\mu F$, $C_{out}=0.1\mu F$; unless otherwise specified.)

| Item | Symbol | Test Conditions | min. | typ. | max. | unit |
|---|-----------------------------|--|--------------|-----------|--------------|-----------------|
| Output Voltage | V_{out} | $T_J=25^{\circ}C$ $10.5V \leq V_{in} \leq 23V$, $5mA \leq I_{out} \leq 1.0A$, $P_D \leq 15W$ | 7.84 7.74 | 8.0 -- | 8.16 8.26 | V |
| Line Regulation | $\Delta V_{REGline}$ | $T_J=25^{\circ}C$ $10.5V \leq V_{in} \leq 25V$ | -- | 6 | 160 | mV |
| | | $11V \leq V_{in} \leq 17V$ | -- | 2.0 | 80 | mV |
| | | $5mA \leq I_{out} \leq 1.5A$ | -- | 12 | 160 | mV |
| Load Regulation | $\Delta V_{REGload}$ | $T_J=25^{\circ}C$ $250mA \leq I_{out} \leq 750mA$ | -- | 4 | 80 | mV |
| Quiescent Current | I_q | $T_J=25^{\circ}C$, $I_{out}=0$ | -- | 4.3 | 8.0 | mA |
| | | $10.5V \leq V_{in} \leq 25V$ | -- | -- | 1.0 | mA |
| Quiescent Current Change | ΔI_q | $5mA \leq I_{out} \leq 1.0A$ | -- | -- | 0.5 | mA |
| Output Noise Voltage | V_n | $T_a=25^{\circ}C$, $10Hz \leq f \leq 100KHz$ | -- | 52 | -- | μV |
| Ripple Rejection Ratio | RR | $f=120Hz$ | 56 | 72 | -- | dB |
| Voltage Drop | V_{drop} | $I_{out}=1.0A$, $T_J=25^{\circ}C$ | -- | 2.0 | -- | V |
| Output Resistance | R_{out} | $f=1KHz$ | -- | 16 | -- | m Ω |
| Output Short Circuit Current | I_{os} | $T_J=25^{\circ}C$ | -- | 450 | -- | mA |
| Peak Output Current | $I_{o peak}$ | $T_J=25^{\circ}C$ | -- | 2.2 | -- | A |
| Temperature Coefficient of Output Voltage | $\Delta V_{out}/\Delta T_J$ | $I_{out}=5mA$, $0^{\circ}C \leq T_J \leq 125^{\circ}C$ | -- | -1.8 | -- | mV/ $^{\circ}C$ |

C.5 Datasheet LM7908

UTCLM79XX LINEAR INTEGRATED CIRCUIT

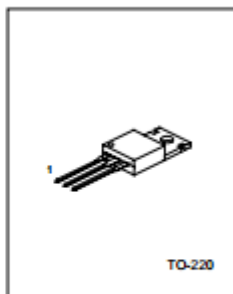
3 TERMINAL 1A NEGATIVE VOLTAGE REGULATOR

DESCRIPTION

The UTC LM7900 series of three-terminal negative regulators are available in TO-220 package and with several fixed output voltage, making them useful in a wide range of application. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible.

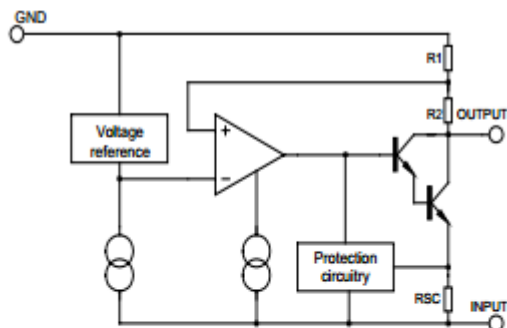
FEATURES

- *Output current up to 1A
- *5V, 6V, 8V, 12V, 15V, 18V, 24V output voltage available
- *Thermal overload protection
- *Short circuit protection



1:GND 2:Input 3:Output

BLOCK DIAGRAM



UTC UNISONIC TECHNOLOGIES CO., LTD. 1

UTCLM79XX LINEAR INTEGRATED CIRCUIT

UTC7908 ELECTRICAL CHARACTERISTICS

(Refer to test circuits, $0 < T_J < 125^\circ\text{C}$, $I_o = 500\text{mA}$, $V_i = -14\text{V}$, $C_i = 2.2\mu\text{F}$, $C_o = 1\mu\text{F}$, unless otherwise specified)

| PARAMETER | SYMBOL | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|--------------------------|-------------------------|---|-------|------|-------|----------------------|
| Output voltage | V_o | $T_J = 25^\circ\text{C}$ | -7.68 | -8.0 | -8.32 | V |
| | | $5.0\text{mA} < I_o < 1.0\text{A}$, $P_o < 15\text{W}$ | | | | |
| | | $V_i = -10.5\text{V}$ to -23V | -7.60 | | -8.40 | V |
| Line regulation | ΔV_o | $T_J = 25^\circ\text{C}$, $V_i = -10.5\text{V}$ to -25V | | 10 | 100 | mV |
| | | $T_J = 25^\circ\text{C}$, $V_i = -11.5\text{V}$ to -17V | | 5 | 80 | mV |
| Load regulation | ΔV_o | $T_J = 25^\circ\text{C}$, $I_o = 5.0\text{mA}$ to 1.5A | | 12 | 160 | mV |
| | | $T_J = 25^\circ\text{C}$, $I_o = 250\text{mA}$ to 750mA | | 4 | 80 | mV |
| Quiescent current | I_q | $T_J = 25^\circ\text{C}$ | | 3 | 6 | mA |
| Quiescent current change | ΔI_q | $I_o = 5\text{mA}$ to 1.0A | | 0.05 | 0.5 | mA |
| | | $V_i = -11.5\text{V}$ to -25V | | 0.1 | 1.0 | mA |
| Output voltage drift | $\Delta V_o / \Delta T$ | $I_o = 5\text{mA}$ | | -0.6 | | mV/ $^\circ\text{C}$ |
| Output noise voltage | V_n | $f = 10\text{Hz}$ to 100kHz , $T_a = 25^\circ\text{C}$ | | 175 | | μV |
| Ripple rejection | RR | $f = 120\text{Hz}$, $V_i = -11.5\text{V}$ to -21.5V | 54 | 60 | | dB |
| Dropout voltage | V_o | $I_o = 1.0\text{A}$, $T_J = 25^\circ\text{C}$ | | 2 | | V |
| Short circuit current | I_{sc} | $V_i = -35\text{V}$, $T_a = 25^\circ\text{C}$ | | 300 | | mA |
| peak current | I_{pk} | $T_J = 25^\circ\text{C}$ | | 2.2 | | A |

RIWAYAT HIDUP PENULIS



Nama : Aristawati Dhiyan Widodo
TTL : Surabaya, 22 Juli 1997
Jenis Kelamin : Perempuan
Agama : Islam
Alamat Rumah : Perum Bluru Permai FC-
18, Sidoarjo
Telp/HP : 085732264643
E-mail : ristadhyn@gmail.com
Hobi : Menonton Film

RIWAYAT PENDIDIKAN

- 2001 – 2003 : TK Trisula
- 2003 – 2009 : SD Negeri Sidoklumpuk
- 2009 – 2012 : SMP Negeri 6 Sidoarjo
- 2012 – 2015 : SMA Muhammadiyah 2 Sidoarjo
- 2015 – sekarang : Bidang Studi Elektro Industri Kerjasama ITS-Disnakertransduk, Program D3 Teknik Elektro, ITS

PENGALAMAN KERJA

- Kerja Praktek di PUSVETMA (Juli - Agustus 2016)
- Kerja Praktek di PT. Campina Ice Cream Industry (Juni - Juli 2017)

PENGALAMAN ORGANISASI

- Sekretaris Departemen Sosial Masyarakat (SOSMAS) Himpunan Departemen Teknik Elektro Otomasi 2017-2018

-----Halaman ini sengaja dikosongkan-----